NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**PHYSICALLY BASED MODELING AND SIMULATION OF A SHIP IN OPEN WATER 3-D VIRTUAL ENVIRONMENT**

by

David Leonardo Lencastre Sicuro

September 2003

| | |
|---|---|
| Thesis Advisors: | Xiaoping Yun |
| | Fotis Papoulias |
| Second Reader: | Joseph Sullivan |

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | *Form Approved OMB No. 0704-0188* |
|---|---|---|---|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503. | | | |
| **1. AGENCY USE ONLY *(Leave blank)*** | **2. REPORT DATE** September 2003 | **3. REPORT TYPE AND DATES COVERED** Master's Thesis | |
| **4. TITLE AND SUBTITLE**: Physically Based Modeling and Simulation of a Ship in Open Water 3-D Virtual Environment | | | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** David Leonardo Lencastre Sicuro | | | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | | | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | | | **10. SPONSORING/MONITORING AGENCY REPORT NUMBER** |
| **11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. | | | |
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release; distribution is unlimited | | | **12b. DISTRIBUTION CODE** |

**13. ABSTRACT *(maximum 200 words)***

This thesis addresses the development of a physically based modeling simulator for a ship in a 3-D virtual environment to be used in naval tactical training systems. The objective is to develop a computer simulation program in which physical models are implemented in order to achieve a realistic representation of a ship in a virtual environment considering its physical features in the presence of environment conditions including waves, ocean current, wind, fog and day/night issues. The simulator was developed by integrating five marine models with a virtual ocean environment created with a visual simulation builder tool. The marine models include a maneuvering model, a wave model, a wind model and an ocean current model. The numerical results from another complex wave model were also combined using linear interpolation to increase the realism level of the simulator. The result of this thesis shows that the integration of multiple models from different sources is a feasible approach to meet the application requirements. The result also indicates that the use of the interpolation technique to take advantage of complex models yields a simulator with acceptable level of realism while imposing very low computational load in the application program.

| **14. SUBJECT TERMS** Physically Based Modeling, 3-D Virtual Environment, Ship Maneuvering Simulator, Environment Disturbances | | | **15. NUMBER OF PAGES** 111 |
|---|---|---|---|
| | | | **16. PRICE CODE** |
| **17. SECURITY CLASSIFICATION OF REPORT** Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified | **20. LIMITATION OF ABSTRACT** UL |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**This thesis done in cooperation with the MOVES Institute**
**Approved for public release; distribution is unlimited**

# PHYSICALLY BASED MODELING AND SIMULATION OF A SHIP IN OPEN WATER 3-D VIRTUAL ENVIRONMENT

David Leonardo Lencastre Sicuro
Technologist Senior, Brazilian Navy
B.S., Rio de Janeiro Federal University, 1983

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

**and**

**MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS AND SIMULATION**

from the

**NAVAL POSTGRADUATE SCHOOL**
**September 2003**

Author: David Leonardo Lencastre Sicuro

Approved by: Xiaoping Yun
Thesis Advisor

Fotis A. Papoulias
Thesis Advisor

Joseph A. Sullivan
Second Reader

John P. Powers
Chairman, Department of Electrical and Computer Engineering

Rudolph P. Darken
Chairman, Department of Modeling, Virtual Environments and Simulation

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

This thesis addresses the development of a physically based modeling simulator for a ship in a 3-D virtual environment to be used in naval tactical training systems. The objective is to develop a computer simulation program in which physical models are implemented in order to achieve a realistic representation of a ship in a virtual environment considering its physical features in the presence of environment conditions including waves, ocean current, wind, fog and day/night issues. The simulator was developed by integrating five marine models with a virtual ocean environment created with a visual simulation builder tool. The marine models include a maneuvering model, a wave model, a wind model and an ocean current model. The numerical results from another complex wave model were also combined using linear interpolation to increase the realism level of the simulator. The result of this thesis shows that the integration of multiple models from different sources is a feasible approach to meet the application requirements. The result also indicates that the use of the interpolation technique to take advantage of complex models yields a simulator with acceptable level of realism while imposing very low computational load in the application program.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

First, I would like to thank the Brazilian Navy for the opportunity to take these graduate courses and enjoy this exciting overseas experience.

Next, I would like to thank all the people who helped me to complete this thesis. Thanks to all the Brazilian Navy Officers and U.S. Navy Surface Warfare Officers, especially my colleague LCDR Steven Perchalski and Prof. Rudolph Darken from the MOVES Institute, for their support in the survey phase. I would like to thank all the staff from the Brazilian Navy Research Institute, and especially my friends LCDR J. L. Marins and LCDR J. J. Andrada for all their overseas support and providing resources from Brazil every time I needed. I would like to thank my advisor Prof. Xiaoping Yun from the Department of Electrical and Computer Engineering for all the orientation, dedication and knowledge. I would like to thank my co-advisor Prof. Fotis Papoulias from the Department of Mechanical Engineering for the precious collaboration in the ship modeling study. I would like to especially thank my second reader, Military Instructor CDR Joseph Sullivan from the MOVES Institute, for his dedication and assistance who actually worked as a third advisor in the software development part. I would like to thank all those from the MOVES Institute staff for their help, especially Erik Johnson and Matt Prichard who provided drawing support for the simulation program. Thanks to my editor Nancy Sharrock for fixing all English mistakes during the preparation of this thesis.

Thanks to all my friends from the Brazilian community in Monterey for providing their priceless support and to Monterey itself for being a perfect place to live.

I would especially like to thank my wife Ivana for her dedication, patience, and for bringing peace and love to my life during the hard times. And finally thanks to God, because without His kindness, we cannot do anything.

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

This thesis addresses the development of a physically based modeling simulator for a generic ship in an open water 3-D virtual environment to be used primarily in naval tactical training systems. The study includes marine dynamic models and their integration for the purpose of developing a simulator using virtual reality techniques. The main objective is to develop a computer simulation program in which physical models are implemented in order to achieve a realistic representation of a ship in a virtual environment considering physical features of the ship such as engine, propeller and skull, in the presence of environment conditions including waves, ocean current, wind, fog and day/night issues.

The approach taken in this thesis was to integrate five marine models with a virtual ocean environment created with the support of a visual simulation builder tool called Vega from MultiGen-Paradigm Inc. Vega is an easy-to-use development tool that incorporates modules for marine and special effects. The object of the simulation is a high-speed container ship. The implemented models include a 4-DOF (degrees of freedom) maneuvering dynamic model in surge, sway, roll and yaw, a 1-DOF wave model that adds random behaviors to the roll motion, and a 3-DOF wind model in surge, sway and yaw. A complex 6-DOF wave model was also combined with the other models to add the pitch and heave motions. This complex wave model runs as a separate application and generates numerical tables that were interpolated in real-time to obtain the six motion outputs during the simulation. The fifth model implemented in this study is a 2-DOF ocean current model in surge and sway motions. Some features were added to the simulation program to facilitate the observation of the environment forces and to verify the ship performance. They are a virtual grid on the ocean, a compass, arrows that represent the environment force vectors and a stopwatch.

The principal objective of this thesis was reached. The simulation program worked well and this study showed that the integration of multiple models from different sources is feasible and produces good results, meeting the application requirements. The

use of the interpolation technique to take advantage of complex models worked well, producing very low computational load in the application program with very acceptable results. The use of four dynamic models with relative low complexity produced sufficiently good results, saving processing time for other tasks when running on complex simulation systems. The Vega system showed to be very useful to build complex virtual environment applications in a short time.

# I. INTRODUCTION

## A. BACKGROUND

As a result of the rapid and cheaper development of computer graphics and hardware, the use of virtual environments (VE) simulators has become a necessity. In military applications, particularly, it represents a feasible approach to experiment with new tactics and weapons, in order to interconnect training units worldwide affordably.

This thesis addresses the preliminary development of a physically based modeling simulator for a generic ship in an open water 3-D virtual environment to be used primarily in naval tactical training systems. It could also be useful in many other maritime simulation areas, such as naval damage control simulators, marine vessel simulators for stability analysis and training systems for entering and leaving port. The complexity of the models to be used will depend on the intended applications.

From the point of view of control systems, this thesis presents ship models with four and six degrees of freedom, environment models and interaction between the models.

When considering naval tactical simulators, the main goal is to take advantage of several features of VE techniques to increase the functionality of a naval tactical simulator. It is not only a matter of obtaining better visual presentation using many texture techniques and viewpoints possibilities. The point is that a simulator using VE must have more functionality than a simulator not using VE. In other words, in a VE tactical simulator, the user must have more resources to obtain useful information from the operation scenery to help him make decisions than when utilizing the old fashion simulators.

For example, if a trainee officer, using the console/cage set as a helicopter, could see the image of a maneuvering ship in the ocean from the virtual cockpit, then the trainee officer would be able to infer which type of evasive movement the ship is undertaking by simply observing the ship's wake. This kind of information is what is available to the user in a VE simulator that would allow the user to feel immersed in the operation. In this same sense, a virtual representation of the sea state, weather conditions, the ship rocking, and so forth, would add a new condition to the exercise. The attitudes taken by an officer during good weather and flat sea scenery certainly would be different from the ones taken during a foggy situation or in a hard sea state condition where the visual clues are bad. Further, the pitch and roll movements of the ship in a stormy sea state condition may change some reception features of sensors, such as radars, thereby creating effects that are very difficult to handle, such as the Sea Clutter, the Image Effect, and the Sea Skimmer, for example. It may also impose constraints in some armament usage.

As can be seen, the functionality and the feeling of immersion would be increased due to the intrinsic characteristics of a VE tactical simulator.

Naturally, all sensor and weapon models used in the tactical simulator would have to be concerned with this new input information, and their complexity would certainly increase to support the rocking ship scenario. Therefore, the computational load of the system would also increase. The tradeoff between the model's complexity and processing time will be discussed later in the following chapter.

## B. PROBLEM STATEMENT

The research question of this thesis is how to simulate physical behaviors of a ship in an open-water virtual environment with respect to its structural characteristics and the environment conditions.

This thesis addresses marine dynamic models and their integration for the purpose of developing a VE tactical simulator. More specifically, the objective is to develop a computer simulation program where physical models are implemented in order to achieve a realistic representation of a ship in a virtual environment considering its physical features such as the engine, propeller and skull, in the presence of environment conditions including waves, ocean currents and wind, fog and day/night issues.

The models used in this study are obtained from the literature. Some adaptations are necessary to work with different models from different sources for the same ship. This is a result of the lack of complete information about the ship that was chosen for this simulation from any single source.

## C. THESIS OUTLINE

Following this introduction, Chapter II describes how the objectives of this simulation are obtained in terms of what is more relevant to model in what kind of application. Some considerations about the tradeoff between the complexity of the models and realism complete this chapter. Chapter III offers an overview of the simulation system presenting its main features and the system modules. Chapter IV describes the Virtual Environment Module. The marine module and the marine special effects module provided by the visual simulation system are also presented. Chapter V explains the Ship's Motion Module. A theoretical background in hydrodynamics is first discussed to introduce some notation and concepts that will make it easier to understand the models. Then, each model is described, starting with the Maneuvering Model. Finally, the environment disturbance models including the Wave Model I and II, the Wind Model and the Ocean Current Model are presented. Chapter VI describes the implementation of the simulation program and explains all the principal modules. Chapter VII presents the method to operate the simulation program and Chapter VIII finalizes presenting the conclusions and recommendations for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. OBJECTIVES OF SIMULATION SYSTEM

## A. WHAT TO MODEL?

There are many parameters to be considered in developing a ship simulation system. It is not possible to consider all parameters in this thesis work. A natural question is: what would the users and experts in tactical training consider important modeling parameters in terms of physical characteristics of the ships and the environment that could make the training more realistic and immersive and, consequently, would increase the functionality of the simulator?

In order to answer the initial question and determine the most important modeling parameters in terms of physical characteristics of the ships and the environment, the following approach was taken. First, some interviews were conducted to ascertain the main issues that could be explored to take advantage of the VE usage in the ship simulation. These issues are related to the modeling of behaviors of a ship under various weather/sea state conditions.

Interview techniques are useful for identifying possible areas for more detailed analysis, and interviews are easy to conduct and direct. The unstructured interview can generate interesting points and statistical analysis can be run on the user's answers [1]. The data collected provide information about general rules and principles and is faster than observational techniques. Interview techniques are useful for investigating events that occur infrequently. In addition, the interviews can be recorded for future analysis.

### 1. Survey

Based on the results of the interviews, a survey was created to obtain the opinions from a considerable number of users in order to rank modeling parameters in terms of relevance. The subjects of this survey were surface warfare officers from the Brazilian Navy and U.S. Navy, and some people involved with naval tactical training in Brazil. Their opinions were collected and analyzed in order to prioritize the research efforts of this thesis.

The survey used in this study appears in Appendix A, exactly as presented to the subjects.

### 2. Results and Analysis

The opinions of fifty people (50% from Brazil and 50% from the United States) were taken and analyzed. The analysis method used was the simple summation of the rank values of each option. Since the most relevant option receives the rank value 1 (one), the option with the lowest summation value will be the most relevant. The results are shown in Table 1.

| Relevance | Option | Total |
|---|---|---|
| 1 | acceleration/deceleration rates | 92 |
| 2 | turning rate | 93 |
| 3 | visibility issues (day/night, fog) | 193 |
| 4 | influence of sea state (ship "rocking") | 195 |
| 5 | influence of the wind | 201 |
| 6 | influence of currents | 207 |
| 7 | weight distribution (tilt/trim) | 284 |
| 8 | wake of the ship | 296 |

Table 1. Survey Results.

The results demonstrate that users and experts established three distinct categories of relevance. The two first options had rank values considerably lower than the others. Thus, a realistic ship movement is basic in this kind of simulator. The second group includes the influence of the environment on the ship behavior. The third group, with the least relevant values, is the weight distribution and the wake of the ship.

The option "Other" was used with several secondary suggestions. Most were related to the modeling of engines, such as the treatment of ship damages that could constraint the performance of the ship in a warfare operation. This feature will not be contemplated in this study but it is an interesting and important area for future implementations.

## B. MODEL COMPLEXITY TRADEOFF

Another important issue is the complexity of the models. It is expected that complex models are more expensive in terms of processing time. An important aspect to bear in mind is that models do not have to include all the details of the real world. Models are evaluated by their ability to provide reasonable answers. They do not have to be perfect representations of the real world but they need to reasonably represent those aspects of the real situation that greatly influence the measurements of interest. In this kind of application, there are visual and analytic outputs to evaluate the effectiveness of the simulation system. Therefore, the models need to be simple enough to create a good visual representation contributing to an increase in the sense of immersion and be able to produce outputs that interact well with other simulation models.

In the process of system implementation, it is necessary to adjust the complexity of some algorithms according to the margin of spare time of processing of the system. This kind of problem is very common when the system relates to networked virtual environment applications [2]. Many times the system must be “tuned” to obtain a better time distribution among rendering, computational calculations and communication processes.

In order to achieve good results combining simplicity and realism, an interpolation technique was tried in this simulation, specifically, in the ship’s wave model. This technique allows the use of the results of a very complex wave model in a very cheap computational way, by interpolating the desired parameters from tables previously generated.

This chapter described how the main objectives of this thesis were achieved based on the opinions of expert people. The next chapter provides an overview of the simulation system by describing how it expects to achieve the desired objectives. The principal features of the simulator are then presented and its component modules are discriminated.

THIS PAGE INTENTIONALLY LEFT BLANK

# III. SIMULATION SYSTEM OVERVIEW

## A. MAIN FEATURES

Based on the results of the survey, the objective of this thesis is to build a preliminary simulation block to possibly observe the behavior of a ship under some environmental conditions. The results of this study would be used as a part of a larger simulation system where other capabilities would be included. Therefore, the main feature of this simulation system is to provide the user ship maneuvering capabilities and to allow changes in the environment conditions. A simulation is presented in a virtual scene in a graphic terminal.

The maneuvering set includes the acceleration/deceleration of ship (engine order) and turnings (rudder order). The environmental settings make it possible to define the amount of daylight in the scenario, fog intensity, the sea state level and the wave-heading angle, the direction and intensity of the wind, and the direction and intensity of the ocean current. Weight distribution issues are not included in the scope of this thesis.

The simulation system provides real-time information about the ship's position and orientation, current engine state, shaft speed and rudder angle, direction and intensity of the wind and ocean current vectors, and the wave-heading angle. This information may be textual and/or graphical.

It is possible to vary the observer's position. The options include the following points of view: perspective view, lateral view, fore view, astern view, above view and on-the-bridge view.

## B. SYSTEM MODULES

The simulation system is divided into modules. In general, it is very unusual in this type of simulation to have a single block including all the modeling equations. Especially in this study, where the principal objective was not to model but to use and to integrate existing maneuvering and environmental disturbance models, the modularity was a necessity. Based on this fact, the simulator is composed of two main modules.

The Virtual Environment module holds three sub-modules. The User's Input/Output is in charge of accepting commands that set internal parameters associated with the ship maneuvering and environment related models, and displaying textual information on the screen. The Graphics sub-module is responsible for rendering the virtual reality scene. The Ocean Model and Special Effects sub-module contains the set of equations that represent the ocean dynamics, ship's wake and bow-wave, field vectors such as wind and ocean current, and some special effects such as ship's smoke. This module also treats daylight and fog issues.

The Ship Motion module holds four sub-modules. Each sub-module plays a specific role in the dynamics that contribute to representing the expected ship behavior. The Maneuvering Model sub-module treats the acceleration/deceleration and the turnings inputs. The Wave Model I and II, the Wind Model and the Ocean Current Model sub-modules provide the motions produced by the sea state conditions, wind and ocean currents respectively. Figure 1 shows the block diagram of the system.

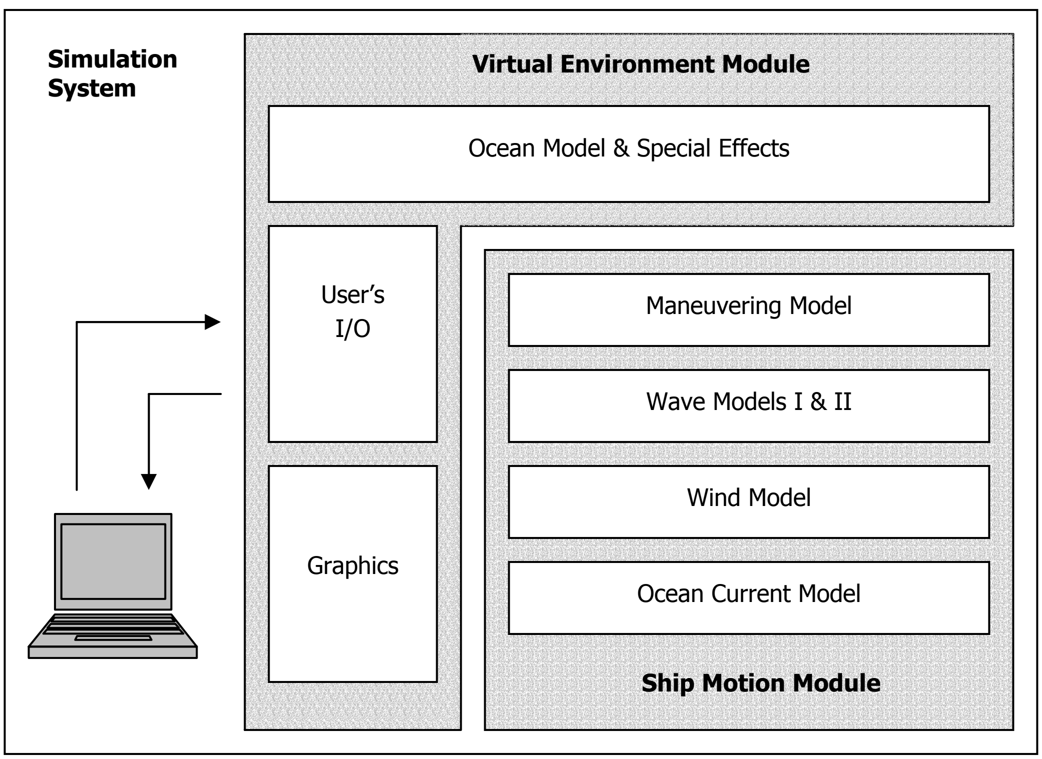



Figure 1. System Modules.

## C. VALIDATION OF THE MODELS

Each model of the Ship Motion module is subjected to a validation process. The validation process has two basic approaches, the analytic and the visual tests. The analytic test is based on measurements of variables of interest that are relevant to a specific test. The measured variables are plotted to make their analysis easier. All measurements were made with the actual simulation program running. A trigger command was implemented in the user's interface to start the recording of pre-defined variables of interest in a textual log file at any time. This log file is in a matrix format in which each row element is related to a measured variable on the simulation test. The first row element is usually reserved for the time variable. Therefore, at each iteration step, a new row is appended, containing the current time value and the respective measurements. When the trigger command is hit a second time, the log file is closed. Then, the log file is imported to the Matlab workspace and the plots are created. The analyses are made through those plots.

The visual test is made by direct observation of the ship behavior in the virtual environment. Basically, this test is based on the common sense about how a ship is supposed to behave under certain conditions.

After the discrimination of the modules, the following step consists of detailing the Virtual Environment module. The next chapter describes the main capabilities of the visual simulation system used in this study.

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. THE VIRTUAL ENVIRONMENT MODULE

There are many tools to help in the development of virtual reality systems due to the high complexity of the applications. To start the rendering using only the original OpenGL functions is practically impossible. The tool used in this thesis is called Vega, developed by MultiGen-Paradigm Inc. Vega has been used in many projects in the Modeling, Virtual Environments and Simulation Department (MOVES) at the Naval Postgraduate School, supporting different virtual reality applications.

## A. THE VEGA SYSTEM

### 1. General Description

Vega is a high performance visual simulation system [4]. It consists primarily of an executable application, several C callable libraries and a graphical user interface. It is modular and extensible in design, and provides users a simple and direct way of interfacing with the target rendering system. The version used in this thesis is 3.7.0. Vega operates on Pentium III platforms or superior, and requires 64 MB of RAM (minimum). A graphics accelerator card with at least 32 MB is required to increase performance. It runs in Windows 2000, Windows NT and SGI Irix.

The goal of Vega is to provide a system to the developer that can be used to construct a simulation application quickly, with the highest performance possible. To accomplish this, Vega includes Lynx, a graphical interface that allows the user to construct the bulk of the application in a point and click environment. Lynx generates an Application Definition File (ADF) that is used to configure the application. Lynx also allows the user to preview the application and make adjustments without leaving the Lynx environment. An ADF may be used to configure a stand-alone executable, or in conjunction with user-supplied code, in order to create complex visual applications.

### 2. The Vega Marine Module

The Vega Marine module provides the capability for Vega to simulate moving ocean waves along with marine special effects. The ocean is simulated as a dynamic component extending to a radius about an observer, and a static component extending further. The dynamic ocean creates a polygonal approximation to a wave field defined by the superposition of 10 sinusoidal components. All components propagate in the same (user-specified) direction. A default set of sinusoidal components is included in the Vega Marine module along with a set of sea states for which the default components are derived using energy spectrum analysis [5].

The virtual waves are created by the Vega Marine module using a sum of 10 sinusoids and a user defined set of non-harmonic frequencies selected to produce a realistic-appearing wave field over a range of sea states. The instantaneous wave elevation is a space-time function computed as follows:

$$\zeta(x, y, t) = T + 0.112 H_s \sum_{i=1}^{10} \cos\left(k_i \left(x \sin\chi + y \cos\chi\right) + \Omega_i \omega_s t + \varphi_i\right) \tag{4.1}$$

where

$T$ = tide, the average height above mean sea level (meters),

$H_s$ = fundamental wave height for the current sea state (meters),

$k_i = (\Omega_i \omega_s)^2 / g$, deep-water dispersion relation of the $i^{th}$ component,

g = gravity acceleration (9.81 m/s$^2$),

$\chi$ = wave direction (0 to $2\pi$),

$\Omega_i$ = non-dimensional angular frequency of the $i^{th}$ component,

$\omega_s = 2\pi/T_s$, angular frequency of sea state,

$T_s$ = the modal period of the sea state,

$t$ = time (s), and

$\varphi_i$ = phase angle of the $i^{th}$ component.

Vega Marine provides application program interface (API) allowing the host application to get and set the parameters associated with the virtual representation of the dynamic ocean. This, in conjunction with a user-furnished dynamics model, allows the host to produce the correlated motion of the ship

### 3. Special Effects

Vega Marine provides many of the marine special effects required by today's sophisticated maritime training applications. It includes astern wakes, bow waves, eddies, buoys, constant-length lines, constant-tension lines, crew overboard, hoisted flags, knuckles, whitecaps, flotsam, foam, log and surf. The models of these special effects are not open to users, but it is possible to adjust some internal parameters to achieve more realistic results. This simulator uses astern wake, bow wave, buoys and eddies.

Furthermore, Vega provides all the resources to control the lightning of the scene, including fog effects. An additional resource that is used to increase the realism of the simulation is the ship's smoke. The smoke follows the wind relative direction, considering the speed of the ship.

### 4. Vega Coordinate System

Vega provides three types of coordinate systems: flat, spherical and elliptical. This work sets Vega in the flat coordinate system, which means that the three axes $X_V$, $Y_V$ and $Z_V$, are oriented as east, north, and up respectively, as depicted in Figure 2.

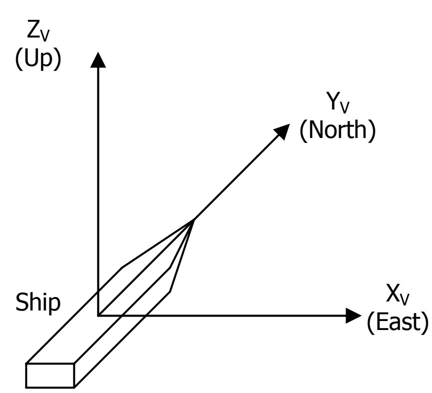



Figure 2. Vega Coordinate System.

Since the coordinate systems used by models may vary, some homogeneous transformations are required to adjust the output parameters to the correct rendering frame. Following the description of each model used, the respective required homogeneous transformation is described.

This chapter described the main features of the Vega system that is the basis of building the Virtual Environment module. Following the description of the system modules, the next chapter describes the Ship Motion module in detail. Initially, a theoretical background is introduced, and then all models related to the motion of the ship are completely explained.

# V. THE SHIP MOTION MODULE

This chapter first presents some important concepts regarding a ship's hydrodynamics in order make it easier to understand the simulation. The chapter then describes the models used in this simulation, starting with the Maneuvering Model. Two wind-generated wave models are used. The Wave Model I is a six degrees-of-freedom (DOF) wave model that works decoupled from the Maneuvering Model. The Wave Model II is coupled to the Maneuvering Model and adds a random component to the roll moment. The description of the Wind and Ocean Current Models completes this chapter.

## A. THEORETICAL BACKGROUND

### 1. Degrees of Freedom and Notations

The motion of marine vehicles in six DOF is considered since six independent coordinates are necessary to determine the position and orientation of a rigid body. The description of each DOF and the respective notation are shown in Table 2.

| DOF index | Description | Forces and moments | Linear and angular veloc. | Positions and Euler angles |
|---|---|---|---|---|
| 1 | Motions in the x-direction (surge) | $X$ | $u$ | $x$ |
| 2 | Motions in the y-direction (sway) | $Y$ | $v$ | $y$ |
| 3 | Motions in the z-direction (heave) | $Z$ | $w$ | $z$ |
| 4 | Rotations in the x-direction (roll) | $K$ | $p$ | $\varphi$ |
| 5 | Rotations in the y-direction (pitch) | $M$ | $q$ | $\theta$ |
| 6 | Rotations in the z-direction (yaw) | $N$ | $r$ | $\psi$ |

Table 2. DOF Description and Notation.

The first three coordinates and their time derivatives correspond to the position and translational motion along the *x*-, *y*- and *z*-axis, while the last three coordinates and time derivatives are used to describe orientation and rotational motion.

Using the above notation, the general motion of a marine vehicle in six DOF can be described by the following vectors (bold variables are vector variables):

$$
\begin{aligned}
&\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T, \boldsymbol{\eta}_2^T]^T, \quad && \boldsymbol{\eta}_1 = [x, y, z]^T, \quad && \boldsymbol{\eta}_2 = [\phi, \theta, \psi]^T, \\
&\boldsymbol{\upsilon} = [\boldsymbol{\upsilon}_1^T, \boldsymbol{\upsilon}_2^T]^T, \quad && \boldsymbol{\upsilon}_1 = [u, v, w]^T, \quad && \boldsymbol{\upsilon}_2 = [p, q, r]^T, \\
&\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T, \boldsymbol{\tau}_2^T]^T, \quad && \boldsymbol{\tau}_1 = [X, Y, Z]^T, \quad && \boldsymbol{\tau}_2 = [K, M, N]^T.
\end{aligned} \tag{5.1}
$$

### 2. Coordinate Frames

When analyzing the motion of marine vehicles in six DOF, it is fitting to define two coordinate frames, as shown in Figure 3, from Ref. 3.

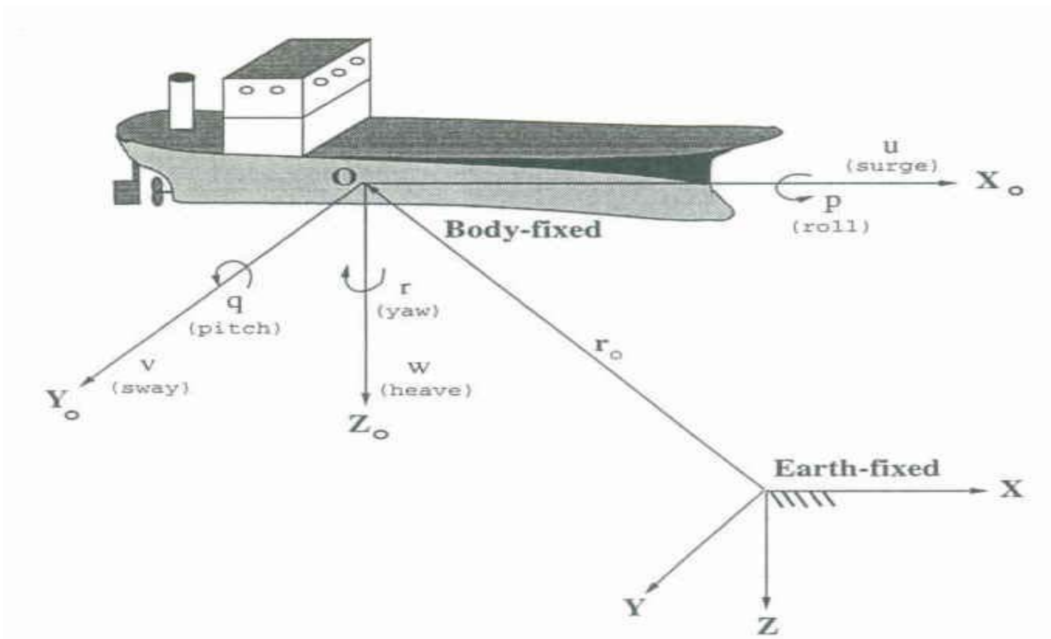



Figure 3. Maneuvering Model Coordinate System.

The moving coordinate frame $X_0\ Y_0\ Z_0$ is fixed to the vehicle and is called the body-fixed reference frame. The origin O of this frame is usually chosen to coincide with the center of gravity (CG) of the ship. For marine vehicles, the body axes $X_0$, $Y_0$ and $Z_0$ coincide with the principal axes of inertia. The motion of the body-fixed frame is described relative to an inertial reference frame $X\ Y\ Z$.

### 3. Ship Motion Equations

Representing the motion equations in the Cartesian system of coordinates (body-fixed reference frame) and defining $x_G$, $y_G$ and $z_G$ as the position of the ship's CG, the well known motion equations of a rigid body [3] are giving by the following:

$$X = m[\dot{u} + qw - rv + x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(rp + \dot{q})] \tag{5.2}$$

$$Y = m[\dot{v} + ru - pw - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] \tag{5.3}$$

$$Z = m[\dot{w} + pv - qu - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] \tag{5.4}$$

$$K = I_X\dot{p} + (I_Z - I_y)qr + m[y_G(\dot{w} + pv - qu) - z_G(\dot{u} + ru - pw)] \tag{5.5}$$

$$M = I_y\dot{q} + (I_x - I_Z)rp + m[z_G(\dot{u} + qw - rv) - x_G(\dot{w} + pv - qu)] \tag{5.6}$$

$$N = I_Z\dot{r} + (I_y - I_x)pq + m[x_G(\dot{v} + ru - pw) - y_G(\dot{u} + qw - rv)] \tag{5.7}$$

where $m$ is the ship's mass; $u, v, w, \dot{u}, \dot{v}, \dot{w}$ represent the linear velocities and accelerations in the $x_0$, $y_0$ and $z_0$ directions; and $p, q, r, \dot{p}, \dot{q}, \dot{r}$ represent the angular velocities and accelerations related to the axes $x_0$, $y_0$ and $z_0$. $I_X$, $I_Y$ and $I_Z$ are the inertia tensors related to same axes of the body-fixed frame. The forces and moments $X$, $Y$, $Z$, $K$, $M$ and $N$ represent the resultants of all external actions over the ship's body.

These equations can be expressed in a more compact form, i.e., the vectorial representation of the 6-DOF rigid-body equations of motion equations:

$$M_{RB}\dot{\boldsymbol{\upsilon}} + C_{RB}(\boldsymbol{\upsilon})\boldsymbol{\upsilon} = \boldsymbol{\tau}_{RB} \tag{5.8}$$

where $M_{RB}$ is the inertial matrix, $C_{RB}$ is the matrix of Coriolis and centripetal terms and $\boldsymbol{\tau}_{RB}$ is the vector of forces and moments.

In basic hydrodynamics, it is common to assume that the hydrodynamic forces and moments on a rigid body can be linearly superimposed by considering the radiation-induced forces and diffraction forces [11]. The radiation-induced forces include added mass due to the inertia of the surrounding fluid, the radiation-induced potential damping due to the energy carried away by generated surface waves, and restoring forces due to Archimedes Law (weight and buoyancy).

Therefore, Eq. 5.8 can be augmented, yielding a more complete form to the vectorial representation of the 6-DOF motion equations:

$$M\dot{\mathbf{\upsilon}} + C(\mathbf{\upsilon})\mathbf{\upsilon} + D(\mathbf{\upsilon})\mathbf{\upsilon} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{5.9}$$

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\mathbf{\upsilon} \tag{5.10}$$

where $M$ is the inertial matrix (including added mass), $C(\mathbf{\upsilon})$ is the matrix of Coriolis and centripetal terms (including added mass), $D(\mathbf{\upsilon})$ is the damping matrix, $\mathbf{g}(\boldsymbol{\eta})$ is the vector of gravitational forces and moments and $\boldsymbol{\tau}$ is the vector of the propulsion forces and moments.

The concept of added mass is usually misunderstood to be a finite amount of water connected to the vehicle such that the vehicle and the fluid represent a new system with mass larger than the original system. Added (virtual) mass should be understood as pressure-induced forces and moments due to a forced harmonic motion of the body, which is proportional to the acceleration of the body [3].

In Eq. 5.10, $J$ is the transformation matrix used to represent the position and orientation vector $\boldsymbol{\eta}$ in the Earth-fixed frame. Equation 5.10 is rewritten below with all details of $J$:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} c\psi & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta & & 0_{3x3} & \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi & & & \\ -s\theta & c\theta s\phi & c\theta c\phi & & & \\ & & & 1 & s\phi t\theta & c\phi t\theta \\ & 0_{3x3} & & 0 & c\phi & -s\phi \\ & & & 0 & s\phi / c\theta & c\phi / c\theta \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{5.11}$$

where $s = sin$, $c = cos$ and $t = tan$.

### 4. Non-Dimensional Parameters

When using hydrodynamic models, it is often suitable to normalize the ship steering equations of motion such that the model parameters can be treated as constants with respect to the instantaneous speed $U$. There are many normalization forms, but the most commonly used for this kind of application is the Prime-system of SNAME [14]. Table 3 shows the Prime-system factors to be applied depending on the unit of the variables of interest.

| Unit | Factor | Unit | Factor |
|---|---|---|---|
| Length | $L$ | Linear velocity | $U$ |
| Mass | $\rho L^3/2$ | Angular velocity | $U/L$ |
| Inertia moment | $\rho L^5/2$ | Linear acceleration | $U^2/L$ |
| Time | $L/U$ | Angular acceleration | $U^2/L^2$ |
| Area | $L^2$ | Force | $\rho U^2 L^2/2$ |
| Position | $L$ | Moment | $\rho U^2 L^3/2$ |

Table 3. Non-Dimensional Prime-System Factors.

This thesis uses the ship's instantaneous speed $U$, the length $L = L_{pp}$ (between the fore and aft perpendiculars) and the salt-water density $\rho$ as normalization variables. The notation used to indicate non-dimensional variables is the "prime" symbol (′). For example, $v' = v/U$ is the normalized (non-dimensional) form of the velocity $v$.

### 5. Hydrodynamics Derivatives

An important step in the development of the maneuvering models is to expand the forces and moments in Taylor's Series. In this way, the nonlinear terms show up as powers of the independent variables, establishing a polynomial equation. To use the expansion, the function and its derivatives need to be continuous and finite in the region of values of the variables under consideration. The precision of the model depends upon the point where the series is truncated. Most models expand the series up to the third power. Some terms can be neglected due to its low significance or because they are null due to the symmetric geometry of the ship. The equation below shows an example of the sway force Y in this form of representation:

$$\begin{aligned} Y = {} & Y_v v + Y_r r + Y_p p + Y_\phi \phi + Y_{vvv} v^3 + Y_{rrr} r^3 + Y_{vvr} v^2 r + Y_{vrr} v r^2 \\ & + Y_{vv\phi} v^2 \phi + Y_{v\phi\phi} v \phi^2 + Y_{rr\phi} r^2 \phi + Y_{r\phi\phi} r \phi^2 \end{aligned} \tag{5.12}$$

where the terms that characterize the hydrodynamic derivatives are represented as

$Y_\phi = \left(\dfrac{\partial Y}{\partial \phi}\right)_{\phi=\phi_0}$, $Y_{rrr} = \left(\dfrac{\partial^3 Y}{\partial^3 r}\right)_{r=r_0}$, $Y_{vvr} = \left(\dfrac{\partial^3 Y}{\partial^2 v \partial r}\right)_{v=v_0, r=r_0}$, and so forth, evaluated at equilibrium conditions. The initial condition of motion equilibrium is chosen as straight ahead motion at a constant speed [14]. These terms are also known as hydrodynamic coefficients.

A large number of experimental methods are available to determine forces and moments associated with variations in linear and angular velocity and acceleration. Typical facilities are the rotation arm, the free oscillator, the forced oscillator, the curved-flow tunnel, the curved models in a straight flow facility and the Planar Motion Mechanism (PMM) technique. Nevertheless, it is difficult to determine all the hydrodynamic coefficients for an ocean vehicle. It is necessary to know these coefficients with reasonable accuracy to obtain a good model of the vehicle [3]. Thus, it is important to notice that the quality of the models relies on experimental databases.

## 6. The Environment Disturbances

Expanding the nonlinear dynamic equation of motion (Eq. 5.9), it becomes:

$$[M_{RB} + M_A]\dot{\boldsymbol{\upsilon}} + [C_{RB}(\boldsymbol{\upsilon}) + C_A(\boldsymbol{\upsilon})]\boldsymbol{\upsilon} + [D_P(\boldsymbol{\upsilon}) + D_V(\boldsymbol{\upsilon})]\boldsymbol{\upsilon} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{5.13}$$

where $M_{RB}$ and $M_A$ are the mass and added mass matrices, $C_{RB}$ and $C_A$ are the Coriolis and centripetal terms matrices for the rigid-body and added mass, and $D_P$ and $D_V$ are the radiation-induced potential damping and viscous damping matrices, respectively.

Applying linearization about zero angular velocity and $u = u_0$, $v = v_0$ and $w = w_0$, the Coriolis and centripetal terms collapse to zero and Eq. 5.13 can be written as follows:

$$[M_{RB} + M_A]\dot{\boldsymbol{\upsilon}} + [N_P + N_V]\boldsymbol{\upsilon} + G\boldsymbol{\eta} = \boldsymbol{\tau} \tag{5.14}$$

where $N_P = \left.\frac{\partial[D_P(\upsilon)\upsilon]}{\partial\upsilon}\right|_{\upsilon=\upsilon_0}$, $N_V = \left.\frac{\partial[D_V(\upsilon)\upsilon]}{\partial\upsilon}\right|_{\upsilon=\upsilon_0}$ and $G = \left.\frac{\partial g(\eta)}{\partial\eta}\right|_{\eta=\eta_0}$ are three constant matrices. The principle of superposition, one of the properties of linear systems, suggests that the environment disturbances can be added to the right hand side of Eq. 5.14 to yield:

$$[M_{RB} + M_A]\dot{\boldsymbol{\upsilon}} + [N_P + N_V]\boldsymbol{\upsilon} + G\boldsymbol{\eta} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{current}\,. \tag{5.15}$$

Let the ocean current velocity vector be defined by $\boldsymbol{\upsilon}_\mathbf{c} = [u_c, v_c, w_c, 0, 0, 0]^T$, where the three last fluid velocity components are zero, assuming irrotational fluid. Assuming also that the vehicle is neutrally buoyant, the mass is homogenously distributed and $\boldsymbol{\upsilon}_\mathbf{r} = \boldsymbol{\upsilon} - \boldsymbol{\upsilon}_\mathbf{c}$ can be interpreted as the relative velocity vector, the linear equations of motion can be combined to yield the following:

$$[M_{RB} + M_A]\dot{\boldsymbol{\upsilon}}_\mathbf{r} + [N_P + N_V]\boldsymbol{\upsilon}_\mathbf{r} + G\boldsymbol{\eta} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{wind} . \tag{5.16}$$

An extension to the nonlinear case proposed by Fossen [16] considers that the following approximation (Eq. 5.17) can be used by ships in the case of treating slowly varying currents in terms of the relative velocity:

$$[M_{RB} + M_A]\dot{\boldsymbol{\upsilon}}_\mathbf{r} + [C_{RB}(\boldsymbol{\upsilon}_\mathbf{r}) + C_A(\boldsymbol{\upsilon}_\mathbf{r})]\boldsymbol{\upsilon}_\mathbf{r} + D(\boldsymbol{\upsilon}_\mathbf{r})\boldsymbol{\upsilon}_\mathbf{r} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{wind} . \tag{5.17}$$

## B. THE MANEUVERING MODEL

An important decision in this study is concerned with the type of ships that would be used. The choice was based on the availability of pre-developed dynamic models and the associated environmental disturbance models. It is preferable that the ship belongs to a very generic class of merchant ships, so that most of the needed information is not classified.

Searching in the literature, a very suitable ship model was found that was written in the Matlab language. It is a 4-DOF nonlinear rolling coupled steering model for a high-speed container ship [3]. In general, for such surface ships, it is common to find models that neglect the heave, pitch and roll modes under the assumption that these motion variables are small. Fortunately, this model includes the roll motion variable that renders the final results more realistic.

The principal limitation of this model is that it only allows forward engine orders. It not only constrains backward maneuvers but also impedes some crash stop maneuvers.

### 1. Container Ship Main Features

The container ship is described by the set of data in Table 4.

| Description | | Symbol | Value |
|---|---|---|---|
| Length | | $L$ | 175.00 m |
| Breadth | | $B$ | 25.40 m |
| Draft | fore | $d_F$ | 8.00 m |
| | aft | $d_A$ | 9.00 m |
| | mean | $d$ | 8.50 m |
| Displacement volume | | $\eta$ | 21,222.00 m$^3$ |
| Height from keel to transverse metacenter | | $KM$ | 10.39 m |
| Height from keel to center of buoyancy | | $KB$ | 4.6154 m |
| Block coefficient | | $C_B$ | 0.559 |
| Rudder Area | | $A_R$ | 33.0376 m$^2$ |
| Aspect Ratio | | $\Lambda$ | 1.8219 |
| Propeller Diameter | | $D$ | 6.533 m |

Table 4. Container Ship Main Features.

### 2. Nonlinear Equations of Motion

A mathematical model for a single-screw high-speed container ship in surge, sway, roll and yaw was presented by Son and Nomoto [12]. In this model, the rigid-body dynamics including the contribution from the hydrodynamic added mass derivatives are presented as follows:

$$X' = (m' + m'_x)\dot{u}' - (m' + m'_y)v'r' \tag{5.18}$$

$$Y' = (m' + m'_y)\dot{v}' + (m' + m'_x)u'r' + m'_y\alpha'_y\dot{r}' - m'_y l'_y \dot{p}' \tag{5.19}$$

$$K' = (I'_x + J'_x)\dot{p}' - m'_y l'_y \dot{v}' - m'_x l'_x u'r' + W'\overline{GM'}\phi' \tag{5.20}$$

$$N' = (I'_z + J'_z)\dot{r}' + m'_y\alpha'_y\dot{v}' + x'_G Y' \; . \tag{5.21}$$

All the equations are in the non-dimensional form, denoted by the symbol ($'$). The variables $m'_x$, $m'_y$, $J'_z$ and $J'_x$ denote the added mass and added moment of inertia in the $x$- and $y$-directions about $z$- and $x$-axes, respectively. The center of added mass for $m'_y$ is denoted by $\alpha'_y$ ($x$-coordinate) while $l'_x$ and $l'_y$ are the added mass $z$ coordinates of $m'_x$ and $m'_y$, respectively. Notice that these equations derive from the standard motion equations of a rigid body (Eq. 5.2-7) with heave and pitch neglected.

The metacentric restoring moment in roll was added in the $K'$ expression (Eq. 5.20), where $W'$ is the weight of the water displaced by the ship hull and GM′ is the transverse metacentric height.

The hydrodynamic forces and moments are defined in equations below:

$$X' = X'(u') + (1-t)T'(J) + X'_{vr}v'r' + X'_{vv}v'^2 + X'_{rr}r'^2 + X'_{\phi\phi}\phi'^2 + c_{RX}F'_N \sin\delta' \quad (5.22)$$

$$\begin{aligned} Y' = {} & Y'_v v' + Y'_r r' + Y'_p p' + Y'_\phi \phi' + Y'_{vvv}v'^3 + Y'_{rrr}r'^3 + Y'_{vvr}v'^2 r' + Y'_{vrr}v'r'^2 \\ & + Y'_{vv\phi}v'^2\phi' + Y'_{v\phi\phi}v'\phi'^2 + Y'_{rr\phi}r'^2\phi' + Y'_{r\phi\phi}r'\phi'^2 + (1+a_H)F'_N \cos\delta' \end{aligned} \quad (5.23)$$

$$\begin{aligned} K' = {} & K'_v v' + K'_r r' + K'_p p' + K'_\phi \phi' + K'_{vvv}v'^3 + K'_{rrr}r'^3 + K'_{vvr}v'^2 r' + K'_{vrr}v'r'^2 \\ & + K'_{vv\phi}v'^2\phi' + K'_{v\phi\phi}v'\phi'^2 + K'_{rr\phi}r'^2\phi' + K'_{r\phi\phi}r'\phi'^2 - (1+a_H)z'_R F'_N \cos\delta' \end{aligned} \quad (5.24)$$

$$\begin{aligned} N' = {} & N'_v v' + N'_r r' + N'_p p' + N'_\phi \phi' + N'_{vvv}v'^3 + N'_{rrr}r'^3 + N'_{vvr}v'^2 r' + N'_{vrr}v'r'^2 \\ & + N'_{vv\phi}v'^2\phi' + N'_{v\phi\phi}v'\phi'^2 + N'_{rr\phi}r'^2\phi' + N'_{r\phi\phi}r'\phi'^2 + (x'_R + a_H x'_H)F'_N \cos\delta' \end{aligned} \quad (5.25)$$

where $X'(u)$ is a velocity-dependent damping function, e.g., $X'(u) = X'_{uu}u'^2$.

The hull parameters used in this model are presented and identified in Table 5. All parameters are in their non-dimensional form.

| Parameter | Description | Value |
|---|---|---|
| $m'$ | Mass of the ship | 0.00792 |
| $m'_x$ | Added mass in the x direction | 0.000238 |
| $m'_y$ | Added mass in the y direction | 0.007049 |
| $I'_x$ | Moment of inertia about x axis | 0.0000176 |
| $J'_x$ | Added moment of inertia about x axis | 0.0000034 |
| $I'_z$ | Moment of inertia about z axis | 0.0000176 |
| $J'_z$ | Added moment of inertia about z axis | 0.0000034 |
| $\alpha'_y$ | Center of added mass $m'_y$ (x-coordinate) | 0.05 |
| $l'_x$ | Added mass z-coordinate of $m'_x$ | 0.0313 |
| $l'_y$ | Added mass z-coordinate of $m'_y$ | 0.0313 |
| $X'_{uu}$ | Hydrodynamic derivative $\partial^2 X / \partial u^2$ | –0.0004226 |
| $X'_{vr}$ | Hydrodynamic derivative $\partial^2 X / \partial v \partial r$ | –0.00311 |
| $X'_{vv}$ | Hydrodynamic derivative $\partial^2 X / \partial v^2$ | –0.00386 |

| Parameter | Description | Value |
| --- | --- | --- |
| $X'_{rr}$ | Hydrodynamic derivative $\partial^2 X / \partial r^2$ | 0.00020 |
| $X'_{\varphi\varphi}$ | Hydrodynamic derivative $\partial^2 X / \partial \phi^2$ | –0.00020 |
| $Y'_{v}$ | Hydrodynamic derivative $\partial Y / \partial v$ | –0.0116 |
| $Y'_{r}$ | Hydrodynamic derivative $\partial Y / \partial r$ | 0.00242 |
| $Y'_{p}$ | Hydrodynamic derivative $\partial Y / \partial p$ | 0.0 |
| $Y'_{\varphi}$ | Hydrodynamic derivative $\partial Y / \partial \phi$ | –0.000063 |
| $Y'_{vvv}$ | Hydrodynamic derivative $\partial^3 Y / \partial v^3$ | –0.109 |
| $Y'_{rrr}$ | Hydrodynamic derivative $\partial^3 Y / \partial r^3$ | 0.00177 |
| $Y'_{rvv}$ | Hydrodynamic derivative $\partial^3 Y / \partial r \partial v^2$ | 0.0214 |
| $Y'_{rrv}$ | Hydrodynamic derivative $\partial^3 Y / \partial r^2 \partial v$ | –0.0405 |
| $Y'_{vv\varphi}$ | Hydrodynamic derivative $\partial^3 Y / \partial v^2 \partial \phi$ | 0.04605 |
| $Y'_{v\varphi\varphi}$ | Hydrodynamic derivative $\partial^3 Y / \partial v \partial \phi^2$ | 0.00304 |
| $Y'_{rr\varphi}$ | Hydrodynamic derivative $\partial^3 Y / \partial r^2 \partial \phi$ | 0.009325 |
| $Y'_{r\varphi\varphi}$ | Hydrodynamic derivative $\partial^3 Y / \partial r \partial \phi^2$ | –0.001368 |
| $N'_{v}$ | Hydrodynamic derivative $\partial N / \partial v$ | –0.0038545 |
| $N'_{r}$ | Hydrodynamic derivative $\partial N / \partial r$ | –0.00222 |
| $N'_{p}$ | Hydrodynamic derivative $\partial N / \partial p$ | 0.000213 |
| $N'_{\varphi}$ | Hydrodynamic derivative $\partial N / \partial \phi$ | –0.0001424 |
| $N'_{vvv}$ | Hydrodynamic derivative $\partial^3 N / \partial v^3$ | 0.001492 |
| $N'_{rrr}$ | Hydrodynamic derivative $\partial^3 N / \partial r^3$ | –0.00229 |
| $N'_{rvv}$ | Hydrodynamic derivative $\partial^3 N / \partial r \partial v^2$ | –0.0424 |
| $N'_{rrv}$ | Hydrodynamic derivative $\partial^3 N / \partial r^2 \partial v$ | 0.00156 |
| $N'_{vv\varphi}$ | Hydrodynamic derivative $\partial^3 N / \partial v^2 \partial \phi$ | –0.019058 |
| $N'_{v\varphi\varphi}$ | Hydrodynamic derivative $\partial^3 N / \partial v \partial \phi^2$ | –0.0053766 |
| $N'_{rr\varphi}$ | Hydrodynamic derivative $\partial^3 N / \partial r^2 \partial \phi$ | –0.0038592 |
| $N'_{r\varphi\varphi}$ | Hydrodynamic derivative $\partial^3 N / \partial r \partial \phi^2$ | 0.0024195 |
| $K'_{p}$ | Hydrodynamic derivative $\partial K / \partial p$ | –0.0000075 |
| $K'_{\varphi}$ | Hydrodynamic derivative $\partial K / \partial \phi$ | –0.000021 |
| $K'_{vvv}$ | Hydrodynamic derivative $\partial^3 K / \partial v^3$ | 0.002843 |
| $K'_{rrr}$ | Hydrodynamic derivative $\partial^3 K / \partial r^3$ | –0.0000462 |
| $K'_{rvv}$ | Hydrodynamic derivative $\partial^3 K / \partial r \partial v^2$ | –0.000558 |
| $K'_{rrv}$ | Hydrodynamic derivative $\partial^3 K / \partial r^2 \partial v$ | 0.0010565 |
| $K'_{vv\varphi}$ | Hydrodynamic derivative $\partial^3 K / \partial v^2 \partial \phi$ | –0.0012012 |
| $K'_{v\varphi\varphi}$ | Hydrodynamic derivative $\partial^3 K / \partial v \partial \phi^2$ | –0.0000793 |

| Parameter | Description | Value |
|---|---|---|
| $K'_{rr\varphi}$ | Hydrodynamic derivative $\partial^3 K / \partial r^2 \partial \phi$ | −0.000243 |
| $K'_{r\varphi\varphi}$ | Hydrodynamic derivative $\partial^3 K / \partial r \partial \phi^2$ | 0.00003569 |

Table 5. Hull Parameters in the Maneuvering Model.

The rudder normal force $F_N$ is resolved as follows:

$$F'_N = -\frac{6.13\Lambda}{\Lambda + 2.25}\frac{A_R}{L^2}\left(u'^2_R + v'^2_R\right)\sin\alpha_R \tag{5.26}$$

where $u'_R$ and $v'_R$ are the surge and sway components of the incident flow velocity in the rudder, and $\alpha_R$ is the incidence flow angle, defined by Eqs. 5.27, 5.28 and 5.29, respectively:

$$u'_R = u'_P \varepsilon \sqrt{1 + \frac{8kK_T}{J^2 \pi}} \tag{5.27}$$

$$v'_R = \gamma v' + c_{Rr} r' + c_{Rrrr} r'^3 + c_{Rrrv} r'^2 v' \tag{5.28}$$

$$\alpha_R = \delta + \tan^{-1}\left(\frac{v'_R}{u'_R}\right). \tag{5.29}$$

Equations 5.30 and 5.31 define $J$, the advance number, and $u'_P$, respectively:

$$J = \frac{u'_P U}{nD} \tag{5.30}$$

$$u'_P = \cos v'[(1 - w_p) + \tau\{(v' + x'_p r')^2 + c_{pv} v' + c_{pr} r'\}]. \tag{5.31}$$

The Bilinear Thruster Model is defined by Eq. 5.32, where $T$ is the thrust developed by the propeller and $K_T(J)$ is the thrust coefficient:

$$T = 2\rho D^4 K_T(J)|n|n \tag{5.32}$$

$$K_T(J) = 0.527 - 0.455J. \tag{5.33}$$

The propeller and rudder parameters used in this model are presented and identified in Table 6.

| Parameter | Description | Value |
|---|---|---|
| $t$ | thrust deduction number | 0.175 |
| $w_p$ | wake fraction number | 0.184 |
| $x'_R$ | x-coordinate of point on which the rudder lateral force $Y_\delta$ acts | –0.5 |
| $z'_R$ | z-coordinate of point on which the rudder lateral force $Y_\delta$ acts | –0.5 |
| $x'_p$ | x-coordinate of propeller position | –0.526 |
| $a_H$ | rudder to hull interaction coefficient | 0.237 |
| $x'_H$ | x-coordinate of point on which normal force $F_N$ acts | –0.48 |
| $c_{RX}$ | constant | 0.71 |
| $\tau$ | constant | 1.09 |
| $\varepsilon$ | constant | 0.921 |
| $k$ | constant | 0.631 |
| $\gamma$ | flow rectification coefficient | 0.088 |
| $c_{pv}$ | propeller flow rectification coefficient | 0.0 |
| $c_{pr}$ | propeller flow rectification coefficient | 0.0 |
| $c_{Rr}$ | rudder wake coefficient | –0.156 |
| $c_{Rrrr}$ | Rudder wake coefficient | –0.275 |
| $c_{Rrrv}$ | Rudder wake coefficient | 1.96 |

Table 6. Propeller and Rudder Parameters in the Maneuvering Model.

### 3. The Model Representation

The maneuvering model is basically represented by Eqs. 5.9 and 5.10. Some simplifications were carried out and adjustments were made to fit only 4-DOF. As a result, the model's equations are:

$$M\dot{\boldsymbol{\upsilon}} = \boldsymbol{\tau} \tag{5.34}$$

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\upsilon} \tag{5.35}$$

The matrix of mass, added mass and inertia $M$ is derived from Eqs. 5.18-21 and expressed as

$$M = \begin{bmatrix} m' + m'_x & 0 & 0 & 0 \\ 0 & m' + m'_y & -m'_y l'_y & m'_y \alpha'_y \\ 0 & -m'_y l'_y & I'_x + J'_x & 0 \\ 0 & m'_y \alpha'_y & 0 & I'_z + J'_z \end{bmatrix}. \tag{5.36}$$

Since $M$ is non-singular, the first set of state derivatives can be written as:

$$\begin{bmatrix} \dot{u}' \\ \dot{v}' \\ \dot{p}' \\ \dot{r}' \end{bmatrix} = M^{-1} \begin{bmatrix} X' \\ Y' \\ K' \\ N' \end{bmatrix}. \tag{5.37}$$

In order to simplify the notation and taking advantage of its symmetry, the matrix $M$ is re-written as:

$$M = \begin{bmatrix} m_{11} & 0 & 0 & 0 \\ 0 & m_{22} & m_{32} & m_{42} \\ 0 & m_{32} & m_{33} & 0 \\ 0 & m_{42} & 0 & m_{44} \end{bmatrix}. \tag{5.38}$$

After computing the inverse of matrix $M$, Eq. 5.37 is explicitly written as follows:

$$\begin{bmatrix} \dot{u}' \\ \dot{v}' \\ \dot{p}' \\ \dot{r}' \end{bmatrix} = \begin{bmatrix} (1/m_{11})X' \\ ((m_{33}m_{44}Y' - m_{32}m_{44}K') - m_{42}m_{33}N')/(m_{22}m_{33}m_{44} - m_{32}^2 m_{44} - m_{42}^2 m_{33}) \\ (-m_{32}m_{44}Y' + m_{22}m_{44}K' - m_{42}^2 K' + m_{32}m_{42}N')/(m_{22}m_{33}m_{44} - m_{32}^2 m_{44} - m_{42}^2 m_{33}) \\ (-m_{42}m_{33}Y' + m_{32}m_{42}K' + m_{22}m_{33}N' - m_{32}^2 N')/(m_{22}m_{33}m_{44} - m_{32}^2 m_{44} - m_{42}^2 m_{33}) \end{bmatrix} \tag{5.39}$$

Returning to the dimensional form, the factor $U^2/L$ is applied to the linear velocity derivatives and the factor $U^2/L^2$ is applied to the rotational velocity derivatives, yielding the final form:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} U^2/L & 0 & 0 & 0 \\ 0 & U^2/L & 0 & 0 \\ 0 & 0 & U^2/L^2 & 0 \\ 0 & 0 & 0 & U^2/L^2 \end{bmatrix} \cdot \begin{bmatrix} \dot{u}' \\ \dot{v}' \\ \dot{p}' \\ \dot{r}' \end{bmatrix}. \tag{5.40}$$

From the 4-DOF adjusted Eq 5.35, the other set of state derivatives can be written as below:

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \\ \dot{\phi}' \\ \dot{\psi}' \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi\cos\phi & 0 & 0 \\ \sin\psi & \cos\psi\cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} u' \\ v' \\ p' \\ r' \end{bmatrix}. \tag{5.41}$$

Again, to return to the dimensional form, the factor $U$ is applied to the position derivatives and the factor $U/L$ is applied to the angle derivatives, as shown below:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} U & 0 & 0 & 0 \\ 0 & U & 0 & 0 \\ 0 & 0 & U/L & 0 \\ 0 & 0 & 0 & U/L \end{bmatrix} \cdot \begin{bmatrix} \dot{x}' \\ \dot{y}' \\ \dot{\phi}' \\ \dot{\psi}' \end{bmatrix}. \tag{5.42}$$

### 4. Rudder Saturation and Dynamics

The rudder dynamics is based on a first order model suggested by Van Amerongen to represent the steering machine [15]. This steering machine model is depicted in Figure 4. Generally, the rudder angle and rudder rate limits are typically in the ranges of $-35^{\circ} < \delta_{\max} < 35^{\circ}$ and $2.33^{\circ}/\mathrm{s} < \dot{\delta}_{\max} < 7^{\circ}/\mathrm{s}$, respectively. In this model, according to the characteristics of the container ship, the rudder angle and rudder rate limits were set at 20º deg and 3º/s, respectively. The parameter $\delta_c$ is the commanded rudder angle.

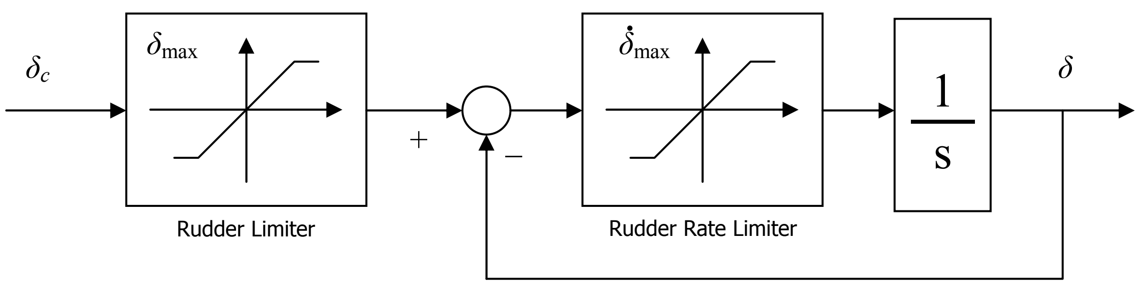



Figure 4. The Steering Machine Model.

### 5. Shaft Speed Saturation and Dynamics

The shaft dynamics used in this model is also a first order system. The behavior of this system depends upon the current shaft speed value. There are two ranges of operation, which determine the value of the parameter $T_m$ to be used. The first one is for speed values lesser or equal to 20 rpm, with $T_m$ being a constant value equal to 18.83. The second is for values greater than 20 rpm, with $T_m$ being a function of $n$, defined by $T_m = 5.65/n$ . Figure 5 shows the shaft model.

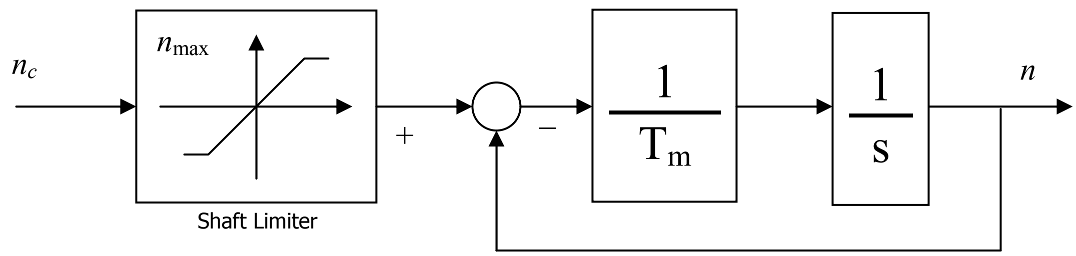



Figure 5. The Shaft Model.

### 6. Interaction with Other Modules

The Maneuvering Model module has basically two input parameters, the commanded shaft speed ($n_c$) and the commanded rudder angle ($\delta_c$). These parameters come from the User's I/O sub-module of the Virtual Environment module. Indeed, the shaft speed order is obtained from the engine order command. Each command corresponds to a shaft speed: Stop = 0 rpm; $^1/_3$ Ahead = 53 rpm; $^2/_3$ Ahead = 107 rpm; and Full Ahead = 160 rpm. The rudder angle order range accepted by this model is from –20º to 20º.

The Maneuvering Model module provides ten state derivative variables: surge and sway linear accelerations, roll and yaw angular accelerations, surge and sway linear velocities, roll and yaw angular velocities, the rudder angular velocity and the shaft angular acceleration. In order to obtain the parameters of interest, it is necessary to integrate these state derivatives over the elapsed time between the renderings of two frames by the Graphics module. This amount of time, $\Delta t$, is measured at each frame-rendering iteration by the Vega's function *vgGetDeltaFrameTime()*. The value is passed to the thread that treats the models to perform the integration. There are many methods to numerically integrate functions [13]. The method used in this thesis was the Euler's integration method due to its simplicity. It worked well in this application because the sampling rate is high (about 60 Hz) compared to the natural frequency of the model. The state derivatives are multiplied by $\Delta t$ and added to their previous values to generate the state variables that are passed to the Virtual Environment module to define the new position and orientation of the ship for the next frame rendering, as below:

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \dot{\mathbf{x}}\Delta t\ . \tag{5.43}$$

The actual values of the rudder angle and the shaft speed are passed to the Virtual Environment module as well in order to be presented on the terminal screen.

The last step before passing the new position and orientation to the Virtual Environment module is to apply a coordinate transformation to convert the results from the Maneuvering Model Coordinate System to Vega Coordinate System. Figure 6 shows the two frames involved in this operation.

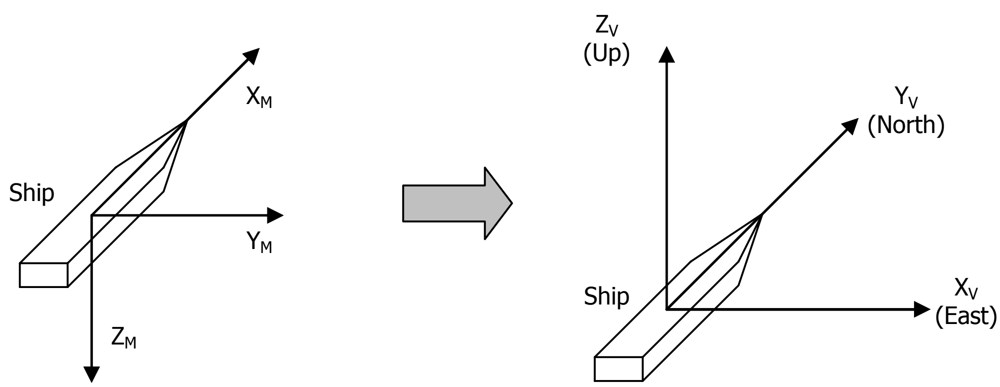



Figure 6. Maneuvering Model to Vega Coordinate System Conversion.

The conversion is achieved by applying a transformation matrix to every position generated by the Maneuvering Model. It must rotate 180 degrees by the $X_M$ axis and then rotate −90 degrees by the $Z_M$ axis. Equation 5.44 shows the homogeneous transformation required.

$$\begin{bmatrix} x_V \\ y_V \\ z_V \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_M \\ y_M \\ z_M \end{bmatrix} = \begin{bmatrix} y_M \\ x_M \\ -z_M \end{bmatrix}. \tag{5.44}$$

The parameters related to rotational motions (roll and yaw) follow the right hand rule in both frames.

The Wave Model II and the Wind Model are coupled to the Maneuvering Model. The interactions among them are discussed in their respective sections.

### 7. Validation of the Maneuvering Model

The idea of this validation section is not to validate the original Matlab code since the model was obtained from a very reliable source. The part that required validation is the C++ implementation of the same model. However, the conversion from Matlab to C++ was very successful. No numerical issues were observed, so that the results obtained in a comparative test were very much the same. Figure 7 shows the surge and sway parameters from the two sources. It is seen that the two output plots (from Matlab original code and from C++ implementation) are perfectly the same, one overlapping the other.

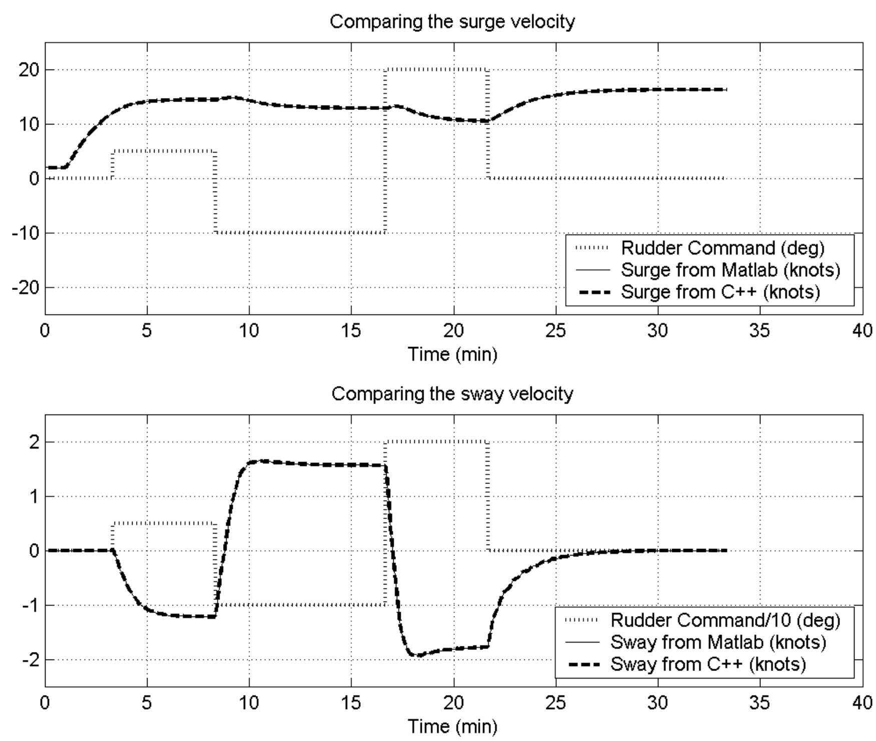



Figure 7. Matlab Code vs. C++ Implementation.

Instead of continuing the presentation of these coincident plots, the rest of this validation section was used to evaluate some performance features of the maneuvering model. At first, from the acceleration/deceleration test, it was verified that the time spent to go from zero speed to the maximum speed (engine order = Full Ahead) was 355 seconds, approximately 6 minutes. The time to do the inverse procedure (engine order = Stop) was greater than 30 minutes. Figure 8a shows the acceleration/deceleration plot.

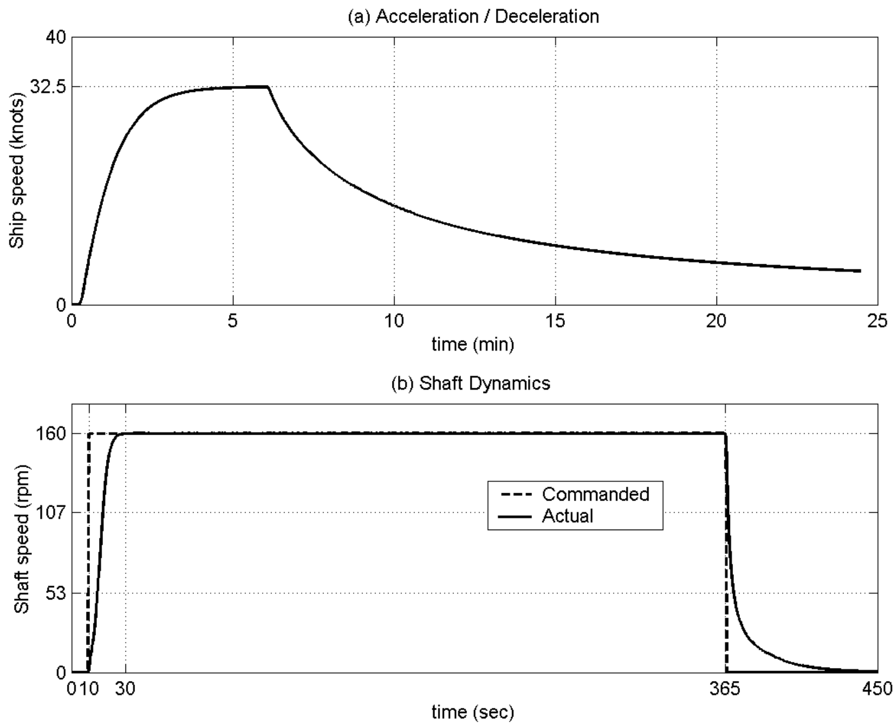



Figure 8. Acceleration / Deceleration and Shaft Dynamics.

Figure 8b demonstrates that the shaft spends approximately 20 s to reach its maximum speed and approximately 85 s to completely stop.

From the rudder dynamics results in Figure 9, it is seen that the time spent to go from the neutral position (0º) to the maximum angle allowed by the model (± 20º) was approximately 12 s. The time to the maximum excursion, from –20º to 20º was approximately 20 s. Since the rudder angle rate did not exceed the limit of 3°/s, i.e., the normal rate for that kind of ship, the result is as expected.

Figure 10 shows the behavior of the surge and sway velocities during maneuvering. As expected, the surge velocity decreases and the sway velocity increases in the opposite direction of the movement during the turnings, according to the inertial reaction. It is also seen from this plot that the surge and sway velocities have a tendency to return to their initial values after the end of all maneuvering.

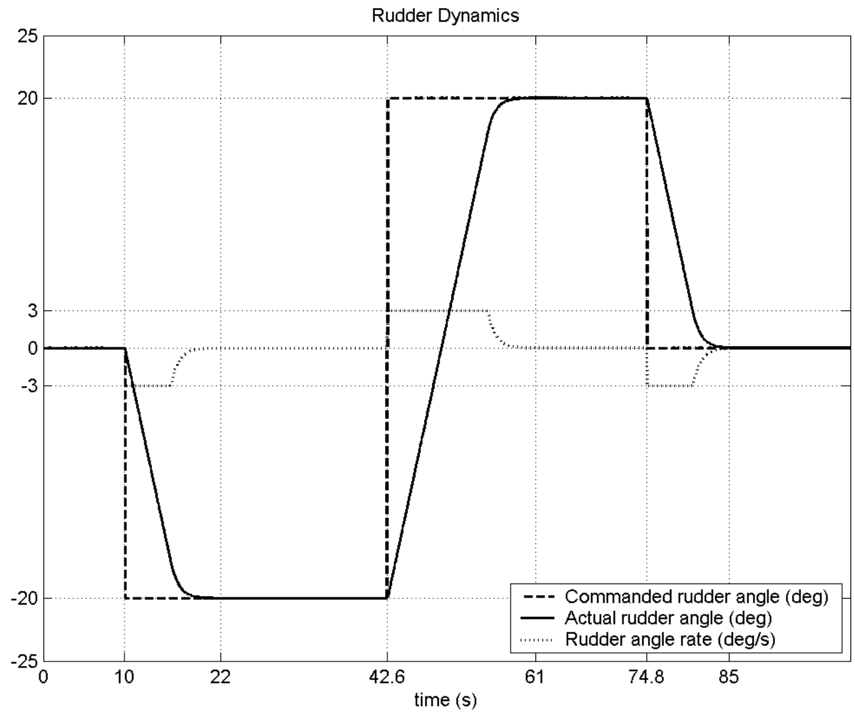



Figure 9. Rudder Dynamics.

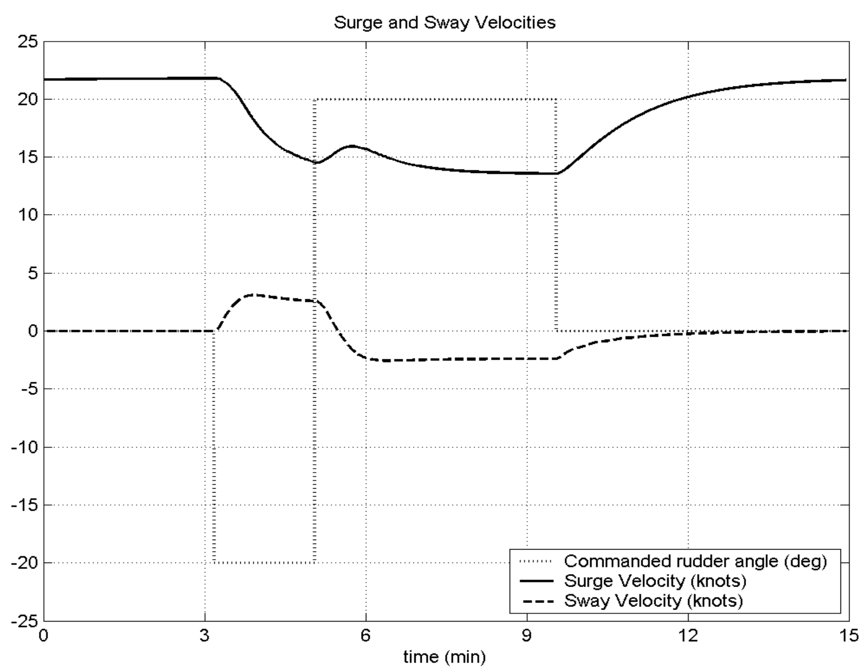



Figure 10. Surge and Sway Velocities.

Other feature observed in this evaluation process was the roll and yaw behaviors during the maneuvers. From Figure 11, it is seen that roll angle has the expected behavior, i.e., the tilt of the ship is in the opposite direction of the turning, according to the inertia. The yaw angle also had the expected performance. The plot shows that, from six to nine minutes, the yaw angle varied approximately 200º, implying a turning rate of 0.667º/s. That rate is perfectly acceptable for that class of ships.

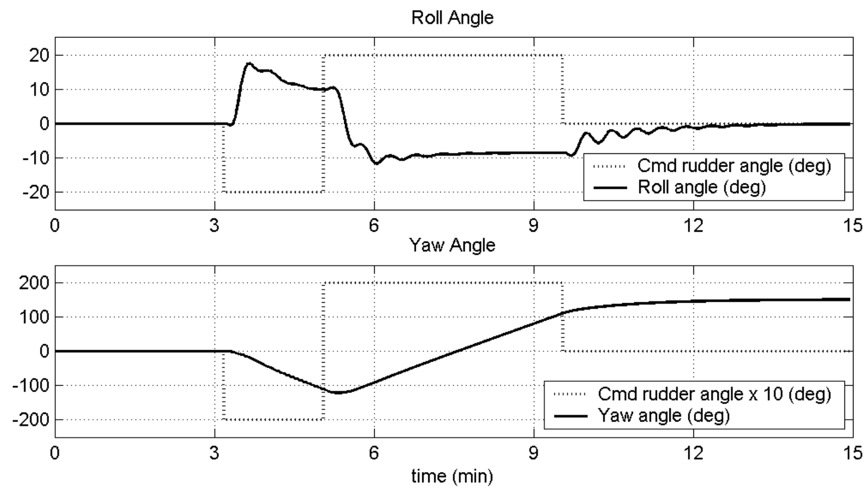



Figure 11. Roll and Yaw Angles.

Two more classic tests were done: the turning circle and the zig-zag maneuvers. The turning circle test was initiated with the ship at a speed of 21.6 knots. At an arbitrary point, a 20º rudder angle turn to starboard started. The parameters evaluated in this test were the steady turning radius (0.245 nm) and the time spent in complete maneuver (approximately 500 seconds). Compared with data from similar ships, those values were considered perfectly valid for this kind of application. Figure 12 shows the turning circle test plot.

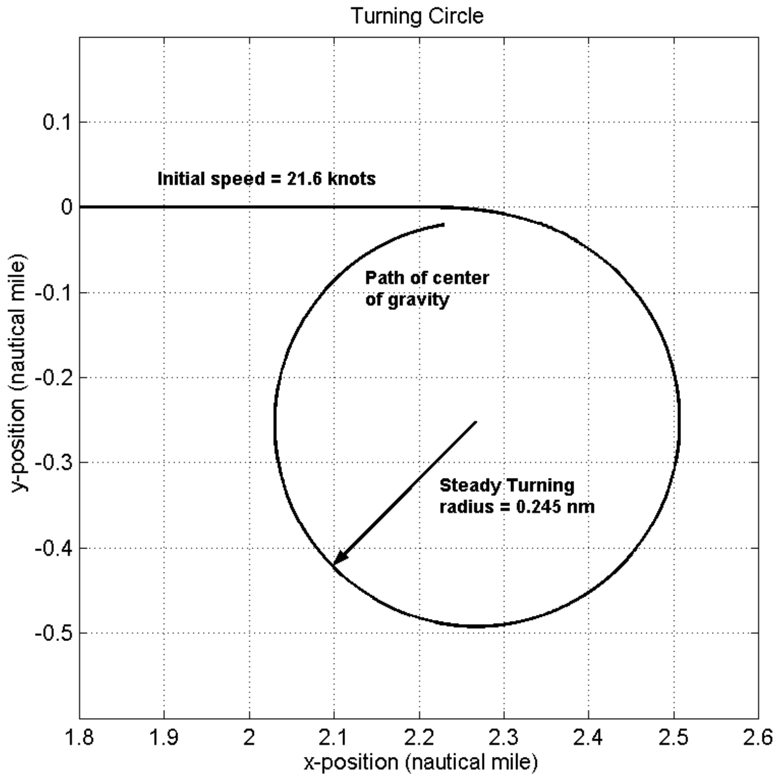



Figure 12. Turning Circle.

The last evaluation test was the Kempf's zig-zag test. This test was first proposed by the German scientist Günther Kempf in 1932 [19]. The zig-zag time response is obtained by commanding the rudder 20º starboard from an initially straight course. The rudder setting is kept constant until the heading is changed 20º, then the rudder is reversed 20º to port. Again, this rudder setting is maintained until the ship heading has reached 20º in the opposite direction. The test continues until a total of five rudder changes have been completed.

The zig-zag test is usually referred to as 20º-20º maneuver, where the first angle refers to the actual rudder settings and the second angle denotes how much the heading angle should change before the rudder is reversed. For large ships, it is recommended to use of a 10º-10º or a 20º-10º maneuver to reduce the time and the space required. In this thesis, the original 20º-20º maneuver was used in order to be possible to compare the results with the available data.

The test started at the constant speed of approximately 11 knots. The elapsed time to reach the first overshoot was 82 s. The average overshoot angle was approximately 6.5º, and the elapsed time to complete the first cycle was 295 s. The final speed was approximately 8 knots. Figure 13 shows the end result of the test.

The results were very compatible with what could be expected from a large container ship behavior. The visual validation corroborated these results. The final evaluation of the maneuvering model was completely satisfactory.

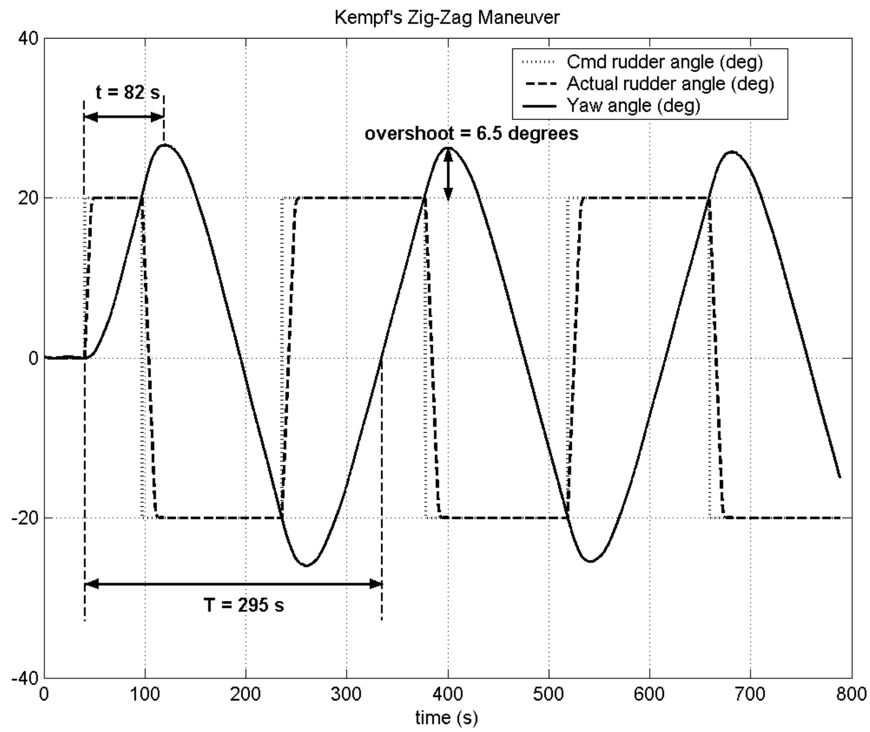



Figure 13. Kempf's Zig-Zag Maneuver.

## C. THE WAVE MODEL I

The Wave Model I is a 6-DOF motion model. It is decoupled from the Maneuvering Model so that its results do not influence ship dynamics. As mentioned previously, in order to achieve good results in terms of realism with a low computational cost, an interpolation technique was implemented in the wave model simulation. This technique allows the use of the results of a complex wave model in a very computationally efficient way, by interpolating the desired parameters from tables previously generated. In order to make this possible, some assumptions were made: 1) the ship's speed varies from 0 to 60 ft/s; 2) the wave heading (WH) varies from 0º to 360º, as shown in Figure 14; 3) the wavelength of the fundamental component of the ocean model varies from 10 to 1460 ft; and 4) the ship has a constant displacement during the entire simulation. Each table provides, for a given (speed, wave heading angle) pair, the values of the output parameters related to the set of pre-defined wavelength values of the fundamental component of the current sea state. The outputs are surge, heave, pitch, sway, roll and yaw. Each of them has amplitude and phase parameters. The computer program described below generates these tables.

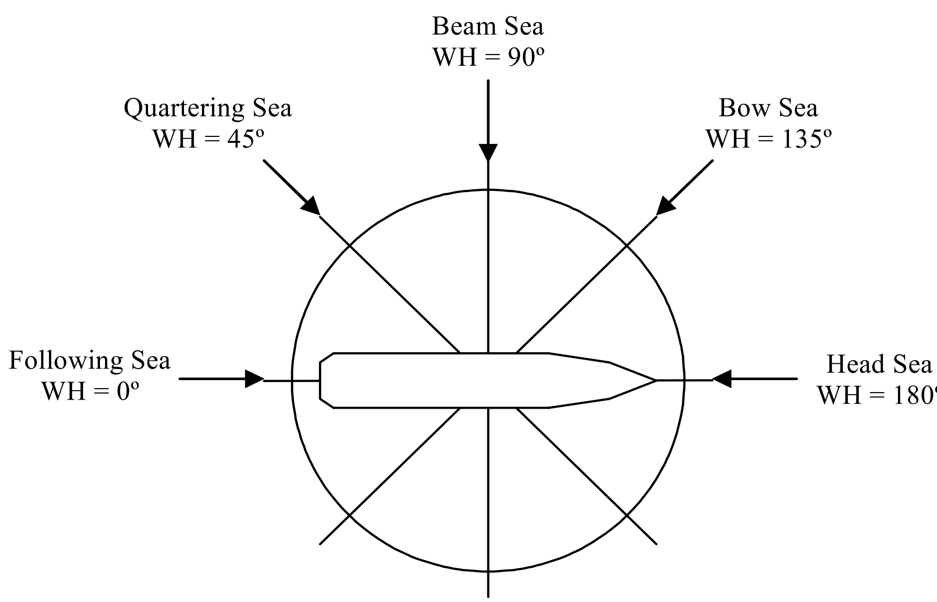



Figure 14. Wave Heading Notation.

### 1. General Description

The computer program called *SHIPMO.BM*, by Robert F. Beck and Armin W. Troesch [6], was developed to predict ship motions in six degrees of freedom and five components of the shear and bending moment distribution. It is based on the strip theory of Salvesen, Tuck and Faltinsen [7] and it was extended by Beck [8] to include the surge degree of freedom. The program computes both regular and irregular wave results.

This program, originally written in FORTRAN, requires two input data files. The first, *SHIPMO.IN*, contains basically three types of information: some internal parameter settings to define the desired operation; information about the ship's structure and the desired output ranges, in terms of ship's speed, wave heading angle and wavelength. The second, G2S1.3, contains terms that are needed in shallow water but not for deep-water calculations, assumed for this thesis, but must be present for the program to run.

The program generates output files in different formats. The file *ADMASS* contains hydrodynamic added mass and damping results. The file *matdata* is an output file with data in Matlab-ready format, while *POT* and *RAO* are files that contain results that could be used in subsequent runs of the program so that it runs faster. The file *VISC* contains viscous roll damping results, and *SHIPMO.OUT* is the main text-based results file.

The *SHIPMO.OUT* file provides non-dimensional values in both vertical and horizontal plane responses. These values are related to surge, heave and pitch parameters in the vertical plane and sway, roll and yaw parameters in the horizontal plane. Each parameter has amplitude (non-dimensional) and phase (degrees). This phase angle represents the difference of phase between each parameter and the encounter wave created by the ship motion based on the Doppler effect. In order to convert the outputs to the dimensional form, the linear motion parameters (surge, sway and heave) are multiplied by the current wave amplitude in the ship's CG. The rotational outputs (pitch, roll and yaw) are multiplied by the same wave amplitude times the wave number. The wave number is defined as $2\pi/\lambda$, where $\lambda$ is the wavelength of the fundamental component of the current sea state.

### 2. Wave Model Coordinate System

The coordinate system adopted by this model is depicted in Figure 15. It has the *X*-axis as the longitudinal axis of the ship (directed from aft to fore), the *Y*-axis as the transverse axis (directed to port) and the *Z*-axis as the vertical axis (directed up).

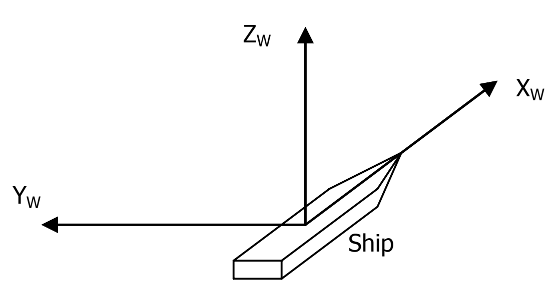



Figure 15. Wave Model Coordinate System.

### 3. Wave Model Tables

In order to obtain the set of tables to execute the interpolation process in the simulation program, two distinct steps are required: 1) run the *SHIPMO.BM* program to generate the output file *SHIPMO.OUT* with the tables, and 2) convert this output file to a C++ type *waveModelTables.h* file. This file contains the tables in two-dimensional array format turning the tables readable to the interpolation program.

#### *a. Obtaining SHIPMO.OUT File*

In order to perform the first task, the wave model's input file *SHIPMO.IN* must be prepared. The internal parameters were set as shown in Table 7.

| Settings | Description |
|---|---|
| 1A = 0 | No even spacing |
| 1B = 0 | Zero weight curve ordinates |
| 1C = 0 | Units on displacement = metric or long tons |
| 1D = 0 | Regular waves model |
| 1E = 1 | Degrees of freedom = both vertical and horizontal plane |
| 1F = 0 | Long crested seas |
| 1G = 1 | Make POT file and no printout on device #7 |
| 1H = 1 | Calculate roll viscous damping |
| 1I = 1 | Non-dimensional output |
| 1J = 0 | Do not do calculations in wave direction |
| 1K = 1 | Generate RAO file |
| 1L = 0 | No shift input offset axis |
| 1N = 0 | No empirical surge damping values do be read in. |
| NS = 13 | Number of input stations |
| BPL = 520 | Ship length (ft) |
| RO = 1.9905 | Water density (slugs / $ft^3$) |
| GRAV = 32.174 | Acceleration of gravity (ft / $s^2$) |
| CDISPL = 0.0 | Auto check of displacement computed value |
| DEPTH = 0.0 | Infinite water depth |
| XBKF = 33.0 | x-location of forward end of bilge keel (ft) (not used) |
| XBKA = –26.0 | x-location of aft end of bilge keel (ft) (not used) |
| BKWTH = 1.0 | Bilge keel width (ft) (not used) |

Table 7. Parameter Settings of *SHIPMO.IN* File (Part I).

Resuming the input settings, the next data is concerned with the ship's body plan. Due to lack of real data, the body plan of a similar ship was used in this simulation, obtained from [10] and depicted in Appendix B. The main structural dimensions as length, breadth, draft and average displacement are very much the same as the actual ship used in the maneuvering model. The skull was segmented in 13 stations, numbered as station 0, ½, 1, 2, 3, 4, 5, 6, 7, 8, 9, 9 ½ and 10. For each station, seven points define a curve in the *Y-Z* plane. This data is used in the two-dimensional potential calculations. The two-dimensional potential, $\psi_k$, is determined by solving the boundary value problem listed in the equation $\nabla^2\psi_k(y,z)=0$ in the fluid domain, where $k$ may be 1, 2, 3 or 4 corresponding to problems for surge, sway, heave or roll, respectively. In this program, the boundary value problem is solved by a source distribution technique [9]. The points used to represent the hull stations are shown in Fig. 16. The points start at keel and work up.

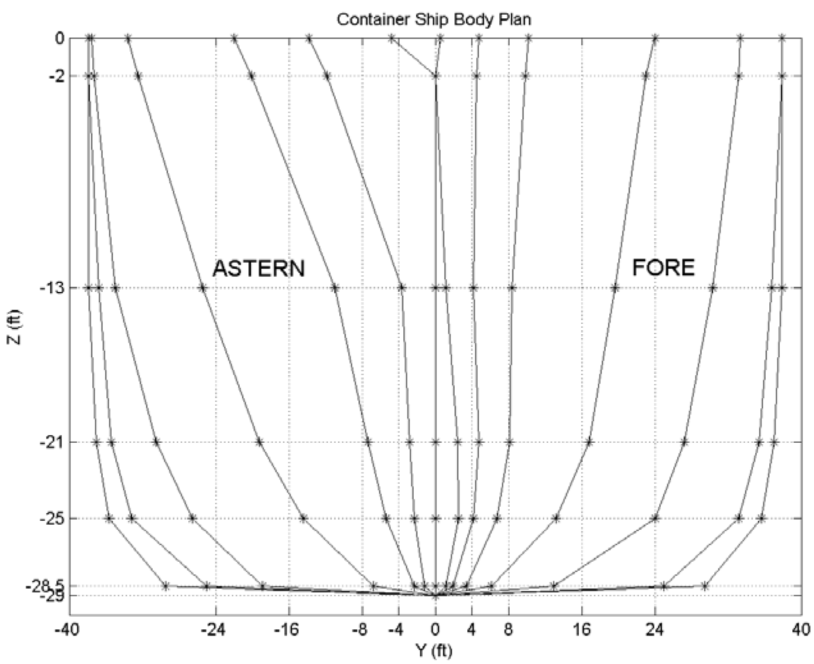



Figure 16. Skull Station Points.

These points are grouped by station sequentially in the input file. For each group, the number of the station, its x-position, the number of points used to describe the curve and the ILID parameter are posted. The parameter ILID = 0 is used to prevent irregular frequencies. The next parameters to be set are ZCG (vertical location of ship center of gravity positive above waterline) and RADGRO (roll radius of gyration about *x*-axis). They are set to 9999 and 0.0, respectively, to indicate that the weight curve computed value will be used.

The last part of the input file used for this simulation addresses the regular wave control. These settings, shown in Table 8, define the number of tables to be generated in the output file. In this case, seven speed steps and 13 wave heading angle steps are used, generating 7x13 = 91 tables. These choices are directly related to the desired level of accuracy in the interpolation process. The smaller the increments, the higher the accuracy, and more tables are created. As long as memory space is not a current concern, depending on the application, more or fewer tables may be created. In other words, if accuracy is requested for a given application, it will not imply that the processing time of the wave model will increase. It will be exactly the same as in applications that do not require much accuracy.

| Settings | Description |
|---|---|
| WA = 1.0 | Continue wave amplitude |
| SWL = 10.0 | Lowest wavelength (ft) |
| BWL = 1460.0 | Biggest wavelength (ft) |
| DELWL = 50.0 | Increment in wavelength (ft) |
| VMIN = 0.0 | Minimum ship speed (ft/s) |
| VMAX = 60.0 | Maximum ship speed (ft/s) |
| DELV = 10.0 | Increment in ship speed (ft/s) |
| WANGI = 0.0 | Initial wave heading angle (degrees) |
| WANGA = 180.0 | Final wave heading angle (degrees) |
| DWANG = 15.0 | Increment in wave heading angle (degrees) |

Table 8. Parameter Settings of *SHIPMO.IN* File (Part II).

The *SHIPMO.IN* file used to generate the tables for this simulation is present in Appendix C.

The output file created by this computer program, *SHIPMO.OUT*, contains information about the ship's structure, stability and the desired tables, one for each (speed/wave heading angle) pair, as a function of wavelength values. Since this is a very large file, only the initial information and the two first sets of tables are shown in Appendix D to provide an idea of its format.

### *b. Converting to waveModelTables.h File*

An auxiliar program called *tableConverter.exe* was implemented to convert the tables to a readable format for the C++ simulation program. This program reads the *SHIPMO.OUT* file as input and generates a float double-precision type, two-dimensional array variable for each table. Each line of this matrix is associated with a wavelength value (from 10 ft to 1460 ft in steps of 50 ft), and the columns are: surge amplitude, surge phase, heave amplitude, heave phase, pitch amplitude, pitch phase, sway amplitude, sway phase, roll amplitude, roll phase, yaw amplitude and yaw phase, in this order. These outputs were named $Amplitude_k$ and $Phase_k$, where $k$ = surge, heave, pitch, sway, roll and yaw. The name of each matrix is automatically created by the program and contains the speed and wave-heading angle to which the table refers. For example, the table named T_Sp40_Wh135 refers to the ship motion parameters related to speed = 40 ft/s and wave heading = 135º.

At the end of the *waveModelTables.h* file, a matrix named TablePointer with the pointers to the first element of each table just converted is created. All these matrices are used to interpolate values in real time during the execution of the simulation.

## 4. Interaction with Other Modules

The four input parameters are needed to produce the 12 outputs that determine the instantaneous ship's position and orientation due the ocean dynamics. The four input parameters are ship's speed, wave heading angle, wavelength of the fundamental wave component and the elevation of the ocean at the ship's CG. The input parameters have two distinct sources, the ship's speed comes from the Maneuvering Model module and the others come from the Virtual Environment Model module.

The instantaneous ship's speed is computed from the current $u$ and $v$ components by using the formula $speed = \sqrt{u^2 + v^2}$ .

The three parameters from the Virtual Environment module are obtained using functions provided by Vega's Application Program Interface (API). The current wave-heading is taken by the *vgGetProp(ocean_calc, VGOCEAN_WAVE_HEADING)* function and the current wavelength is taken by applying the wave period value obtained from the function *vgGetProp(ocean_calc, VGOCEAN_WAVE_PERIOD)* in the formula $wavelength = T_w^2 g / 2\pi$ , where $T_w$ is the wave period and $g$ the gravity acceleration.

Here, an integration issue between the wave model and Vega arose, concerning the wave elevation at ship's CG parameter. Vega's API allows getting current wave elevation values of any point in the virtual sea (*X-Y* plane) but these values are the summation of all ten cosines components. It is not able to obtain the elevation value from a specific component. The wave model needs only the current wave elevation of the fundamental wave component. In order to solve this problem, a second ocean was created in the virtual scenario, 20 meters below the original one. This second ocean, called *ocean_calc*, has the same characteristics of the original one, called *ocean_view*, but contains only one cosine component, i.e., the fundamental wave component. Therefore, the system is now composed by 2 parallel virtual oceans 20 meters apart, but only the *ocean_view* can be seen in the scenario. The *ocean_calc* is used only to get the parameters for calculations. Figure 17 shows a sample of the relationship between the *ocean_view* and the *ocean_calc* wave elevations.

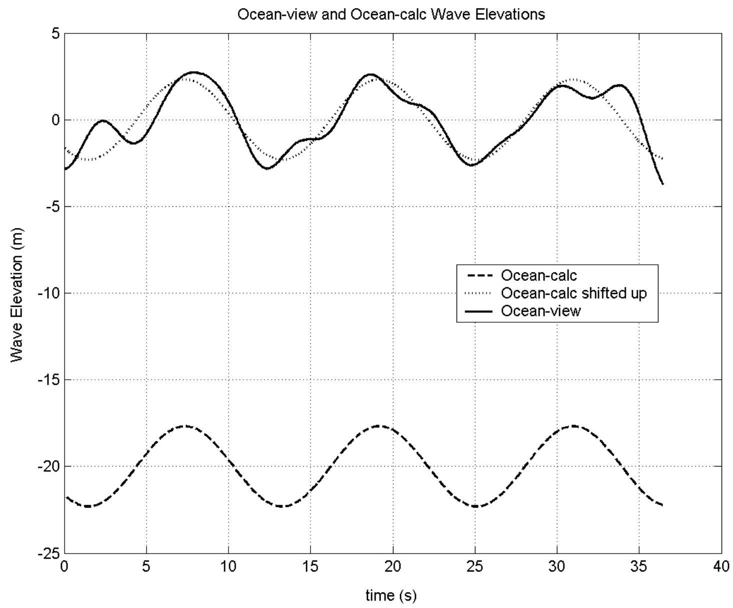



Figure 17. The *ocean_view* and the *ocean_calc* Wave Elevations.

This solution does not imply an important loss in the system performance since the second ocean is not rendered. The major ocean computation time is due to rendering and texturing.

Now that a pure cosine wave in the *ocean_calc*, has been provided, it is needed to determine *where* in the wave the ship's CG is, in order to apply the corresponding phase angle and obtain the correct wave elevation to compute the right dimensional value for each output parameter. It was achieved by taking the difference between two measurements of the wave elevation, in a short distance apart, in the direction of the current wave-heading angle. It is similar to computing the derivative of the wave signal to get the *slope* of the point of interest (ship's CG). Since the maximum wave elevation value is accessible from the current configuration of the ocean, it is possible to compute how far the ship's CG is from the peak of the cosine wave. In other words, this distance corresponds to an offset phase angle ($\alpha$). Figure 18 shows the instantaneous ship's CG position in the wave.

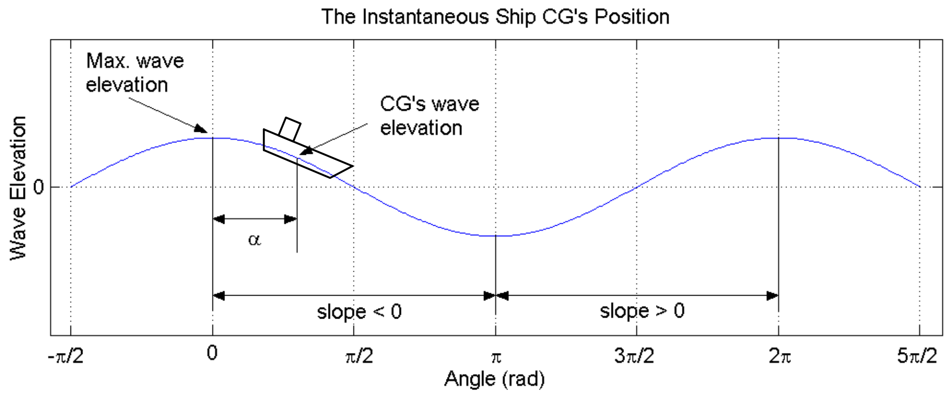



Figure 18. The Instantaneous Ship CG's Position.

The angle α is computed using $\alpha = \cos^{-1}\left(\frac{WaveElev_{CG}}{WaveElev_{\max}}\right)$, and if the slope is positive, $\alpha = 2\pi - \alpha$.

Once α is obtained, the wave elevation related to each output parameter can be calculated as follows:

$$WaveElev_k = WaveElev_{\max} \cdot \cos\left(\alpha + Phase_k\right) \tag{5.45}$$

where $k$ = surge, heave, pitch, sway, roll and yaw. The final values from the wave model for each output parameter are obtained applying the dimensional factors. Equation 5.46 computes the dimensional outputs for the linear motions: surge, sway and heave.

$$output_k = WaveElev_k \cdot Amplitude_k \tag{5.46}$$

where $k$ = surge, sway and heave. Equation 5.47 computes the dimensional outputs for the rotational motions:

$$output_k = WaveElev_k \cdot Amplitude_k \cdot WaveNumber \tag{5.47}$$

where $k$ = pitch, roll and yaw.

Now, it is necessary to apply a homogeneous transformation to convert the results from the Wave Model Coordinate System to Vega Coordinate System. Figure 19 shows the two frames involved in this operation.

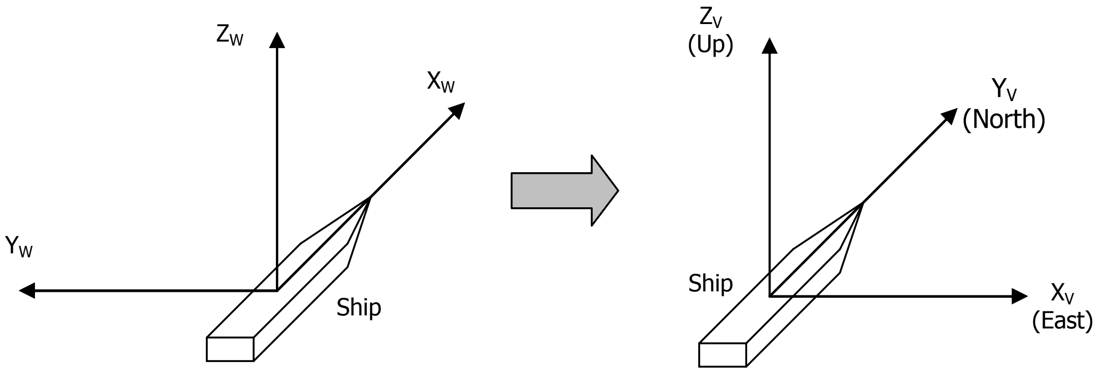



Figure 19. Wave Model to Vega Coordinate System Conversion.

The conversion is carried out by applying a transformation matrix to every position generated by the Wave Model. Now, it is required to rotate –90° by the $Z_W$ axis. Equation 5.48 shows the homogeneous transformation required.

$$\begin{bmatrix} x_V \\ y_V \\ z_V \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} = \begin{bmatrix} -y_W \\ x_W \\ z_W \end{bmatrix}. \tag{5.48}$$

The parameters related to rotational motions (pitch, roll and yaw) follow the right hand rule in both frames.

The ship's motion results generated by the Wave Model module are added to the results obtained by the Maneuvering Model module (only the four degrees-of-freedom in common). Thus, the wave model results do not influence the internal dynamics of the Maneuvering Model. A more coupled integration between the two models was attempted. Due to the non-linear behavior of the Maneuvering Model, the final result became unstable.

### 5. Validation of the Wave Model I

The validation of the results of this model was based on two approaches, analysis of the curves (where it is possible to check any important discrepancies between the interpolated outputs of the model and the expected real situation) and the visual check using the graphic visualization provided by the simulator program. In the first approach, four sets of curves were generated that represent the Wave Model outputs in four ship's speeds: 0, 10.7, 21.6 and 32.5 knots. These speeds correspond to the steady state values of the four engine order regimes: stop, ⅓ ahead, ⅔ ahead and full ahead. For all those cases, the following parameters were used: sea state = 3, dominant wave length = 360 ft, and wave heading varying from 0° to 180° in steps of 45°. Each plot presents the wave elevation (dashed line) at the ship's CG and a specific output parameter. The objective of these plots is to show the amplitudes and phase-shifts between the ocean wave and each output under different wave heading angles.

Figure 20 shows the vertical plane motion parameters (surge, heave and pitch) at speed of 0 knots.

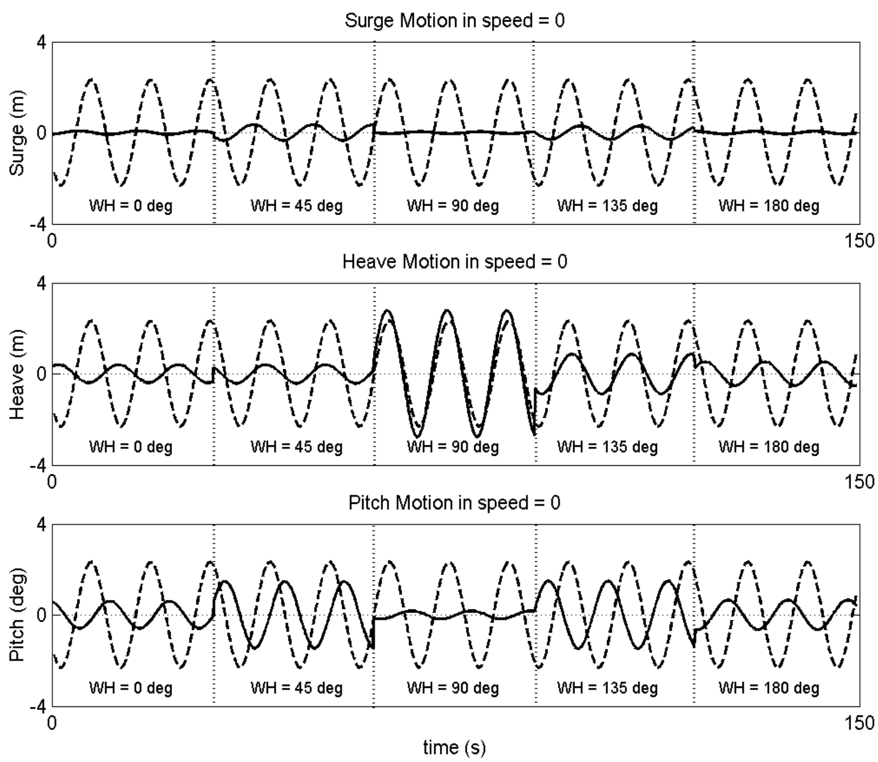



Figure 20. Vertical Plane Motion Output (Speed = 0 Knots).

Observing the plots in Figure 20, it can be seen that the surge motion is not sensitive to the following and head seas, and practically null in the beam sea condition. As expected, the heave motion has a very accentuated behavior in the beam sea and a moderate behavior in the other sea conditions. In the pitch motion, it is seen as an accentuated oscillation in the quartering and bow seas, and a very small response in the beam sea condition, as expected.

Figure 21 shows similar plots to the horizontal plane motion parameters sway, roll and yaw. Notice that this model has response zero in the following and head seas for all motions in the horizontal plane. The sway motion is more accentuated in the beam sea condition, as expected. The yaw motion has a very small response for all wave-heading conditions. All these analytical responses are very reasonable considering the type of the ship. The visual feedback from the graphics animation was very acceptable also, even in high sea state conditions, represented by sea states four to six.

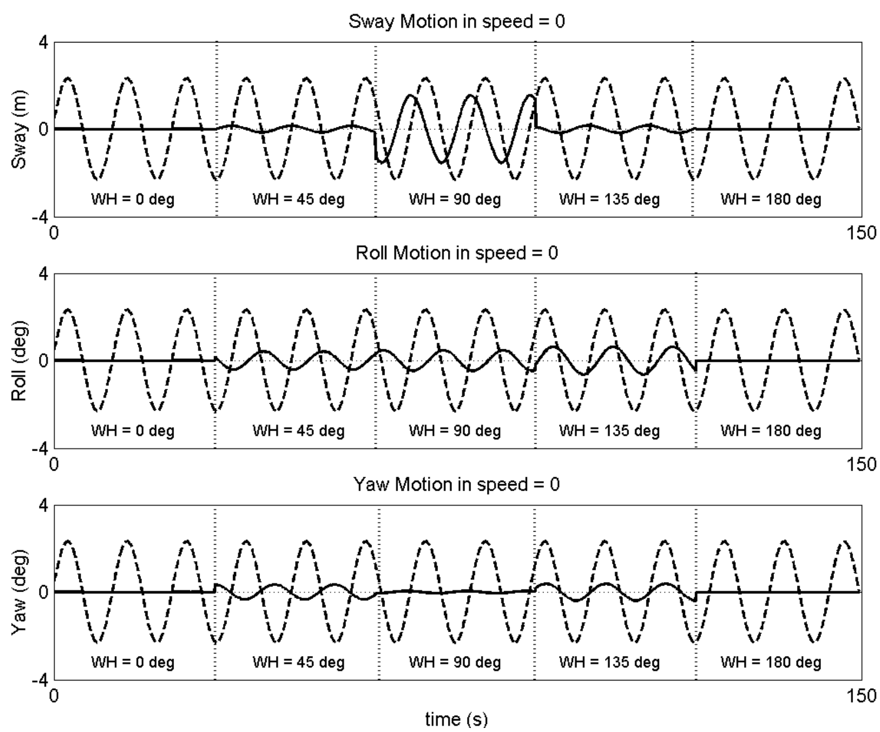



Figure 21. Horizontal Plane Motion Output (Speed = 0 Knots).

Figures 22 and 23 show the vertical and horizontal plane motion outputs at

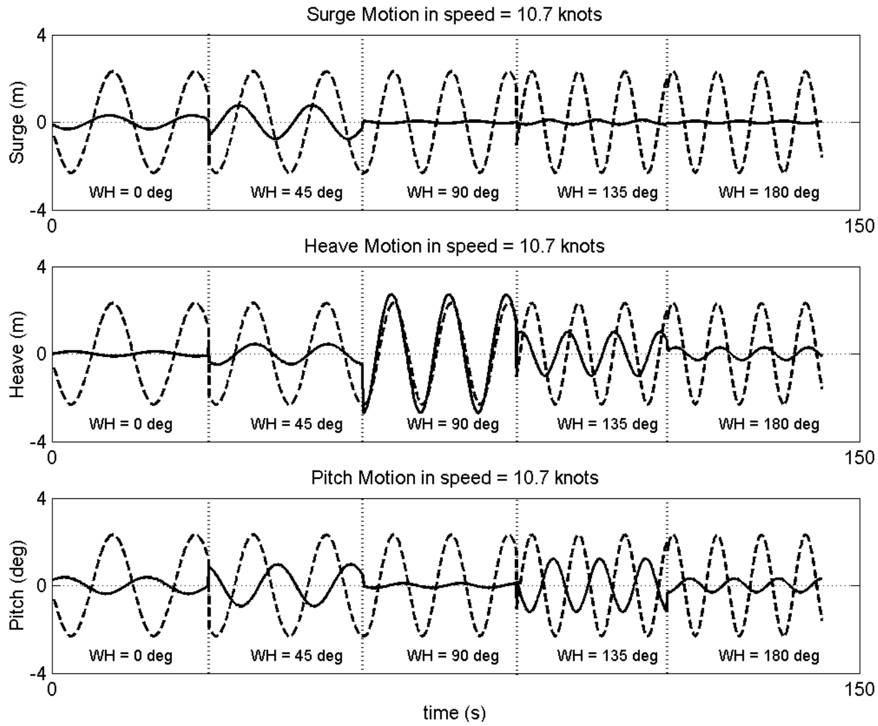



Figure 22. Vertical Plane Motion Output (Speed = 10.7 Knots).

speed of 10.7 knots, respectively.

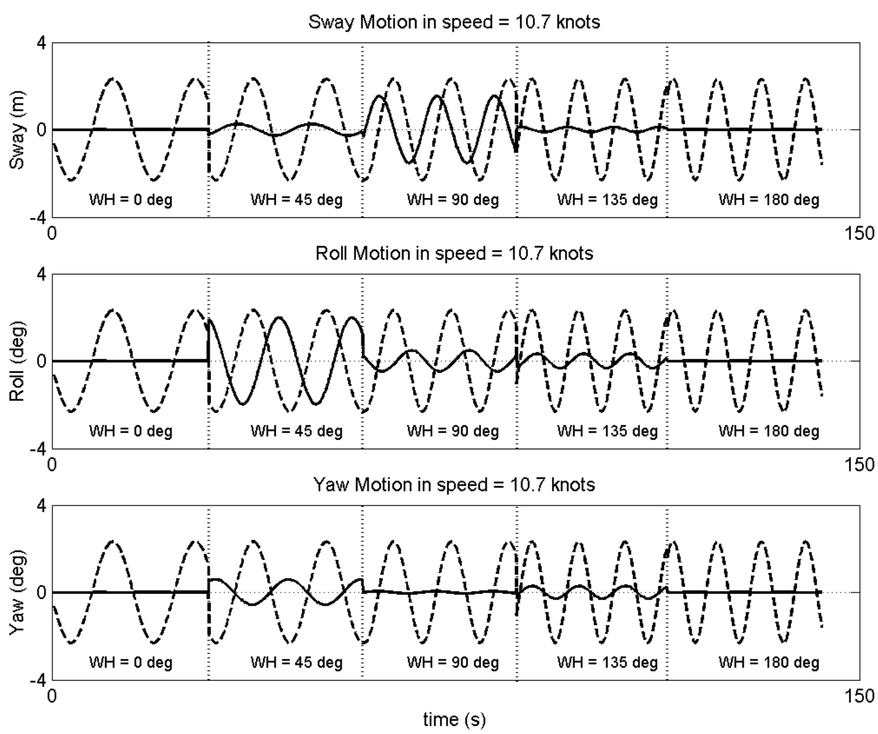



Figure 23. Horizontal Plane Motion Output (Speed = 10.7 Knots).

From the plots of Figures 22 and 23, it is seen that the outputs have reasonable values, which could be attested by the visual check. Also, notice that the wave elevation signal increases in frequency as the wave heading goes to 180 deg because of the Doppler effect.

Figures 24 and 25 show the vertical and horizontal plane motion outputs at speed of 21.6 knots, respectively.

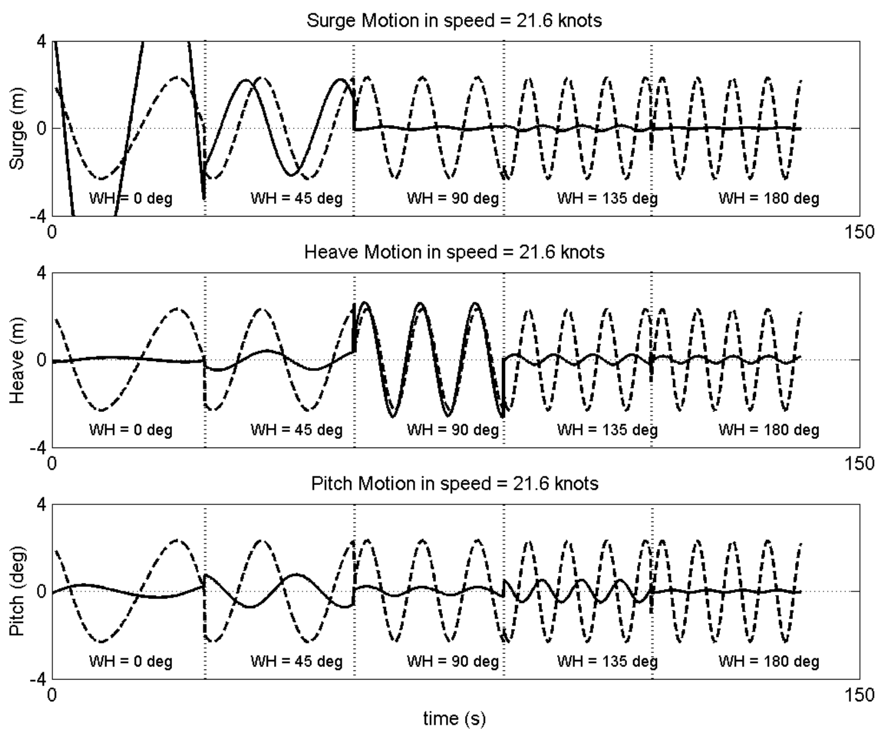



Figure 24. Vertical Plane Motion Output (Speed = 21.6 Knots).

As the speed increases, an unexpected behavior started to occur, especially in the surge and sway motions. It was verified from *SHIPMO.OUT* file tables that, possibly due to some numerical issues, the amplitude values of some parameters became unstable for speed values greater than 10 knots. It was also noticed in the visual check that, even though the plots seem to be correct, the ship did not behave as expected, especially in high seas conditions.

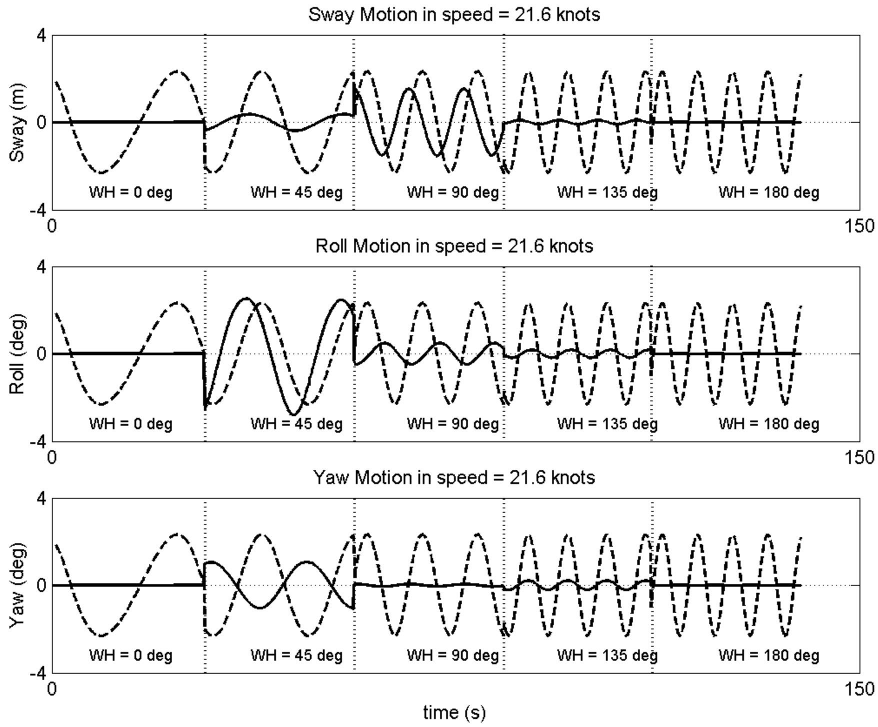



Figure 25. Horizontal Plane Motion Output (Speed = 21.6 Knots).

The plots of Figures 26 and 27 show the problem worsening at the maximum

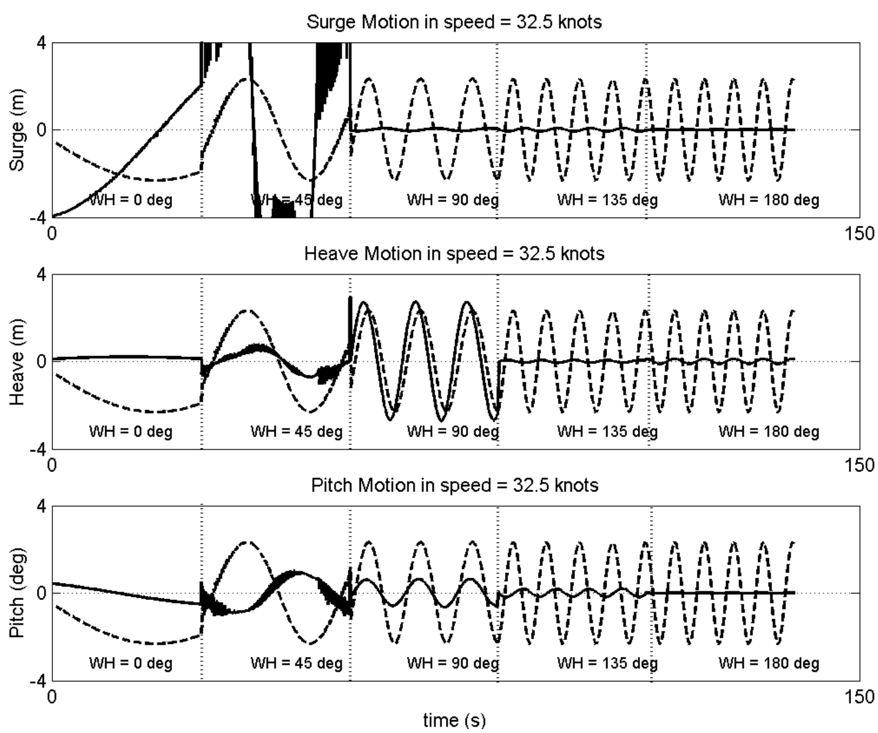



Figure 26. Vertical Plane Motion Output (Speed = 32.5 Knots).

ship's speed of 32.5 knots.

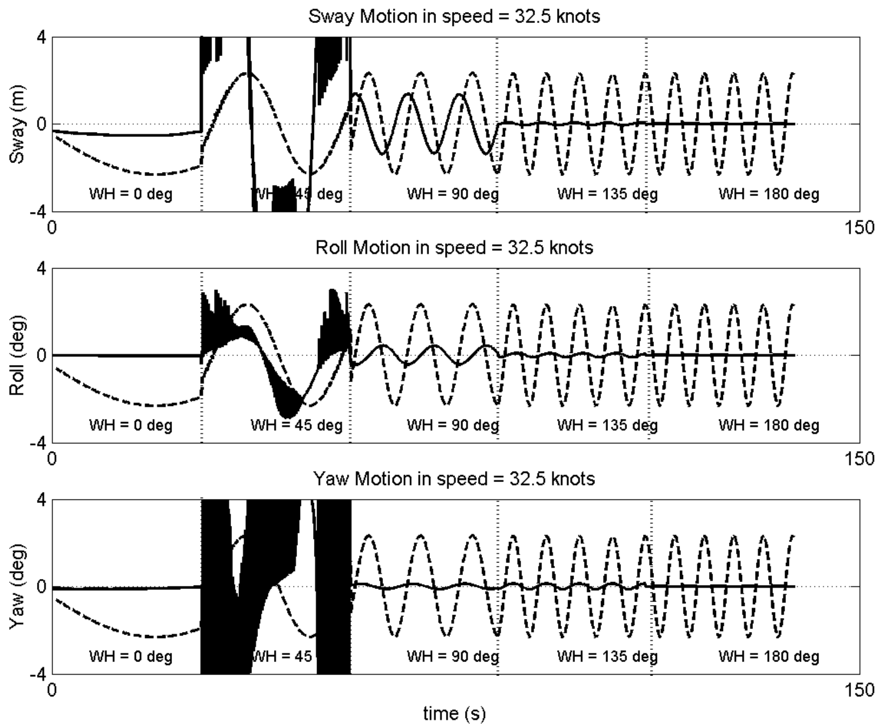



Figure 27. Horizontal Plane Motion Output (Speed = 32.5 Knots).

The implemented solution to avoid this kind of instability was to neglect the surge and sway contributions from this wave model when the speed of the ship is greater than approximately 20 knots.

## D. THE WAVE MODEL II

### 1. Description

The Wave Model II is a wind generated wave model coupled to the Maneuvering Model. It generates an additional moment to the *K* component, contributing to the effect on the roll motion. This is a second order linear wave model. This kind of approximation is usually preferred by ship control systems engineers due to its simplicity and applicability. The first applications were reported by Balchen, Jenssen and Saelid [17]. The model's transfer function is shown below:

$$h(s) = \frac{K_w s}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{5.49}$$

where the gain $K_w$ is defined as $K_w = 2\zeta\omega_0\sigma_w$. The parameter $\zeta$ is the damping coefficient, $\omega_0$ is the dominating wave frequency and $\sigma_w$ is the wave intensity coefficient. The values used for $\zeta$ and $\sigma_w$ were 0.00005 and 1.0, respectively.

The model in the state-space representation is defined by:

$$\begin{bmatrix} \dot{x}_{h1} \\ \dot{x}_{h2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix} \cdot \begin{bmatrix} x_{h1} \\ x_{h2} \end{bmatrix} + \begin{bmatrix} 0 \\ K_w \end{bmatrix} w(t) \tag{5.50}$$

where $w(t)$ is a zero mean white noise process. It introduces a random factor to the ship's roll motion behavior. The state variable $x_{h1}$ is the additional moment $K'_{wave}$ added to Eq. 5.24 according to the principle of superposition.

### 2. Interaction with Other Modules

This model works like a sub-module of the Maneuvering Model module as long as they are coupled. It uses the actual value of the wave frequency from the Virtual Environment module. The white noise process input is internally generated by the mathematical function *randnormal()*. The output parameter $K'_{wave}$ is used by the Maneuvering Model module during the calculations of the forces and moments. The new states $x_{h1}$ and $x_{h2}$ are obtained together with the new states of the Maneuvering Model module as in Eq. 5.43.

### 3. Validation of the Wave Model II

The validation of the Wave Model II was made by repeating the same test used when the Maneuvering Model was validated. The test ran in sea state = 3 and the measurements started with the ship at the speed of 21.6 knots.

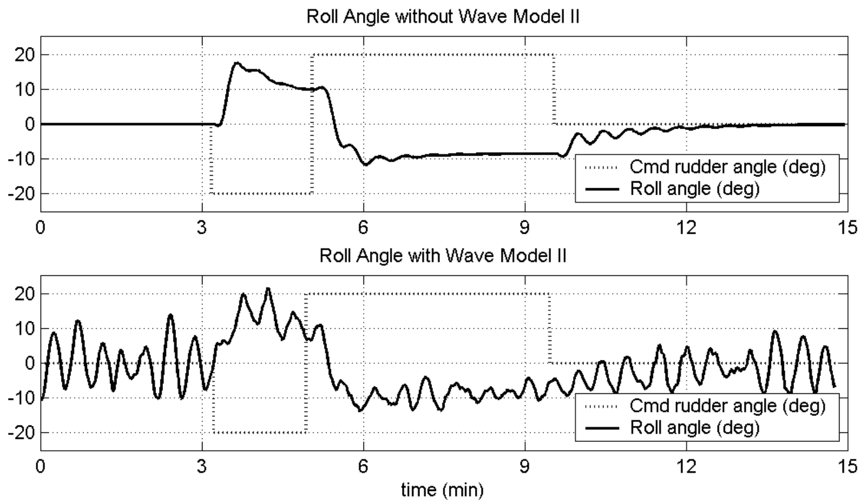



Figure 28. Roll Angle with and without Wave Model II.

The following maneuvers were made: a 20° rudder angle turn to port followed by a −20° rudder angle turn to starboard, and then returning to the neutral rudder position. Figure 28 shows the two test conditions. It is clear the effect of the principle of superposition in the roll motion when the Wave Model II is coupled to the Maneuvering Model. The use of the Wave Model II created a random behavior in the ship motion, making the final result more realistic. The introduction of the $K'_{wave}$ component caused a small disturbance in the Maneuvering Model's yaw motion. However, this disturbance does not produce any unstable behavior in the model and it is not perceptible in the visual check.

## E. THE WIND MODEL

### 1. Description

The resultant wind forces and moments acting on a surface vessel are usually defined in terms of relative wind speed $V_R$ (knots) and relative angle $\gamma_R$ (deg) that are defined by:

$$V_R = \sqrt{u_R^2 + v_R^2} \tag{5.51}$$

$$\gamma_R = \tan^{-1}\left(\frac{v_R}{u_R}\right) \tag{5.52}$$

where the components of $V_R$ in the $x$ and $y$ directions are defined by:

$$u_R = V_w \cos(\gamma) - u + u_c \tag{5.53}$$

$$v_R = V_w \sin(\gamma) - v + v_c\,. \tag{5.54}$$

Here, $u$ and $v$ are the ship velocity components, the $u_c$ and $v_c$ are the ocean current velocity components, $V_w$ is the wind speed while $\gamma = \psi_w - \psi$ is the angle of relative wind of the ship bow.

Typically wind models only treat the force and moments that are directly related to surge, sway and yaw motions, i.e., $\boldsymbol{\tau}_{wind} = [X_{wind}, Y_{wind}, N_{wind}]^T$. The model adopted in this study is one suggested by Isherwood [18], which considers wind resistance of merchant ships. This model defines the wind forces for surge and sway, in Newtons (N), and the wind moment for yaw, in Newton-meters (Nm) as is shown in Eqs. 5.55-57, respectively:

$$X_{wind} = \frac{1}{2} C_X(\gamma_R) \rho_w V_R^2 A_T \tag{5.55}$$

$$Y_{wind} = \frac{1}{2} C_Y(\gamma_R) \rho_w V_R^2 A_L \tag{5.56}$$

$$N_{wind} = \frac{1}{2} C_N(\gamma_R) \rho_w V_R^2 A_L L \tag{5.57}$$

where $C_X$, $C_Y$ and $C_N$ are the force and moment coefficients, $\rho_w$ is the density of the air in kg/m$^3$, $A_T$ and $A_L$ are the transverse and lateral projected areas in m$^2$ and $L$ is the overall length of the ship in meters. Based on these equations, measured data were analyzed by multiple regression techniques in terms of eight parameters: length ($L$), beam ($B$), lateral projected area ($A_L$), transverse projected area ($A_T$), lateral projected area of superstructure ($A_{SS}$), length of perimeter of lateral projection of model excluding waterline and slender bodies such as masts and ventilators ($S$), distance from bow of centroid of lateral projected area ($C$), and number of distinct groups of masts or kingposts seen in lateral projection ($M$). Furthermore, Isherwood [18] found that the data were best fitted to the following equations:

$$C_X = A_0 + A_1 \frac{2A_L}{L^2} + A_2 \frac{2A_T}{B^2} + A_3 \frac{L}{B} + A_4 \frac{S}{L} + A_5 \frac{C}{L} + A_6 M \tag{5.58}$$

$$C_Y = B_0 + B_1 \frac{2A_L}{L^2} + B_2 \frac{2A_T}{B^2} + B_3 \frac{L}{B} + B_4 \frac{S}{L} + B_5 \frac{C}{L} + B_6 \frac{A_{SS}}{A_L} \tag{5.59}$$

$$C_N = C_0 + C_1 \frac{2A_L}{L^2} + C_2 \frac{2A_T}{B^2} + C_3 \frac{L}{B} + C_4 \frac{S}{L} + C_5 \frac{C}{L} \tag{5.60}$$

where the coefficients $A_i$ and $B_i$ ($i = 0\ldots6$) and $C_j$ ($j = 0\ldots5$) come from tables in function of the relative angle $\gamma_R$, that varies from 0 to 180 degrees. These tables were implemented in the simulation program and all coefficients are computed in real time by first order interpolation according to the current relative angle $\gamma_R$.

The values used for the eight parameters related to the container ship of this study are shown in the Table 9.

| **Parameter** | **Value** |
|---|---|
| $L$ | 175 m |
| $B$ | 25.40 m |
| $A_L$ | 1400.0 m$^2$ |
| $A_T$ | 203.3 m$^2$ |
| $A_{SS}$ | 200.0 m$^2$ |
| $S$ | 16 m |
| $C$ | 90 m |
| $M$ | 1 |

Table 9. Ship Parameters for the Wind Model.

In order to be used correctly, the parameters $X_{wind}$, $Y_{wind}$ and $N_{wind}$ were converted to their non-dimensional forms, $X'_{wind}$, $Y'_{wind}$ and $N'_{wind}$, by applying the associated factors described in Table 3, i.e., the force parameters were divided by $\rho U^2 L^2/2$ and the moment parameter was divided by $\rho U^2 L^3/2$.

### 2. Interaction with Other Modules

The wind model works like a sub-module of the Maneuvering Model module as long as they are coupled. The input parameters come from the Environment Model module. They are the $u$ and $v$ components of the relative wind, and the current relative angle $\gamma_R$. The Maneuvering Model module uses the output parameters $X'_{wind}$, $Y'_{wind}$ and $N'_{wind}$ during the calculations of the forces and moments.

### 3. Validation of the Wind Model

The validation of the Wind Model was made through the observation of the ship behavior under several situations. One test is reported here to allow its analysis. The initial conditions were the ship's heading angle = 0° and the wind's heading angle = 270°, i.e., the wind is coming from starboard. The wind intensity was set as 50 knots in order to make the measurements easier in terms of magnitude of the values and the duration of the test. It was observed that the ship tends to be aligned with the wind, i.e., the wind makes the ship's bow move to the opposite direction in the wind. This movement is seen from Figure 29 in the roll and yaw plot. The relative angle $\gamma_R$ starts at π/2 rad (wind coming from starboard) and goes to π rad (head wind). The ship's heading angle starts at 0 rad and goes to π/2 rad (90°), pointing exactly to where the wind is coming from, at the end

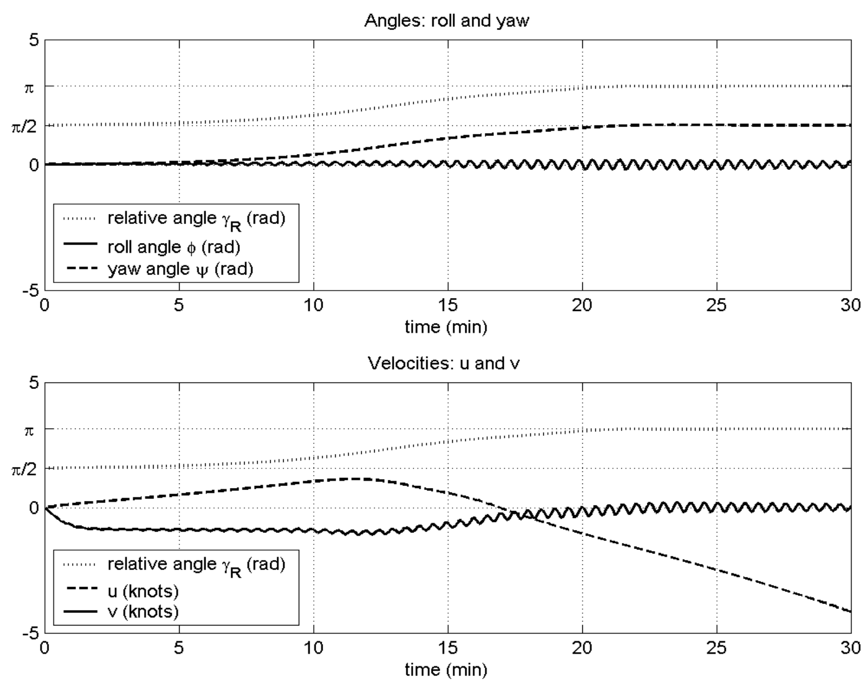



Figure 29. Wind Model Test.

of the simulation. Since the models are coupled, a roll movement was induced too, as seen in the same plot. The amplitude of the roll motion increased as the simulation time increased, but when the ship became aligned with the wind, the roll amplitude started to decrease, as expected. From the velocities plot in the same figure, notice that the ship is rotating until about 18 minutes of simulation and then the ship is pushed backwards by the wind. An oscillated behavior is seen in the $v$ velocity component. It is associated to the induced roll motion.

The observed behavior of the ship during the simulation was as expected and the model was validated.

## F. THE OCEAN CURRENT MODEL

### 1. Description

The ocean current model used in this study is a simple two-dimensional current model. It is commonly used in surface vessel applications. This model considers the local effect of the current, applied to the ship's CG. Therefore, this model has two parameters only, which are the average current speed $V_c$ and direction of the current $\beta$. The two velocity components in the Earth-fixed frame can be written as:

$$u_c^E = V_c \cos\beta \tag{5.61}$$

$$v_c^E = V_c \sin\beta\ . \tag{5.62}$$

Considering the horizontal motion only, those components can easily be converted to the fixed-body frame using the transformation matrix below:

$$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \cdot \begin{bmatrix} u_c^E \\ v_c^E \end{bmatrix} \tag{5.63}$$

where $\psi$ is the heading angle of the ship. Substituting Eqs. 5.61 and 5.62 into Eq. 5.63 finally yields:

$$u_c = V_c \cos(\beta - \psi) \tag{5.64}$$

$$v_c = V_c \sin(\beta - \psi)\ . \tag{5.65}$$

The parameters $V_c$ and $\beta$ can be changed by the user at any time during the simulation.

### 2. Interaction with Other Modules

The ocean current heading angle and velocity parameters are obtained from the Environment Model module. The outputs of the model, the velocity components $u_c$ and $v_c$, are then integrated as in Eq. 5.43, to produce the new $x$ and $y$ position values after the integration interval of time $\Delta t$. The velocity components $u_c$ and $v_c$ also are used to set the input parameters to the Wind Model module as shown in Eqs. 5.53 and 5.54.

The ocean current has a graphic representation in the simulation scenario. The two buoys are presented at the beginning of the simulation, located on both sides of the ship. Their eddies are rendered as a function of the parameters $V_c$ and $\beta$. Far from the initial point of the simulation, the effects of the ocean current are more easily noticed if the ship is just drifting. Otherwise, the user can check the parameter values at any time on the screen in text format.

### 3. Validation of the Ocean Current Model

The validation of this model was very simple. From the initial point of simulation, with the ship in the drifting condition, the ship motion can be easily verified in the direction of the ocean current heading angle, related to the two fixed buoys.

This chapter presented a theoretical discussion about hydrodynamic models and described in detail all the motion models used in this thesis. The following chapter is related to the software implementation. The simulation system is divided into blocks and the software developed in each block is described.

THIS PAGE INTENTIONALLY LEFT BLANK

# VI. THE SIMULATION PROGRAM

This chapter shows the structure of the simulation program. All program components and the relationship between them are described.

## A. GENERAL DESCRIPTION

The main purpose of this simulation program was to provide the user a virtual laboratory to test the maneuvering of a marine vessel under a pre-defined set of environment disturbances through the observation of its behavior. The program provides resources to help the user during the simulation. Three arrows that represent each direction vector for wind, ocean current and wave heading, a 0.1 nautical-mile grid, a compass and a stopwatch can be activated by the user at any time during the simulation.

The program was built in the Microsoft Visual Studio 6.0 environment. It is a multi-thread C++ application and the threads communicate via shared memory. The simulation program runs as an application layer over the Vega system, using Vega's API (application program interface). Figure 30 shows the structure of the simulation program.

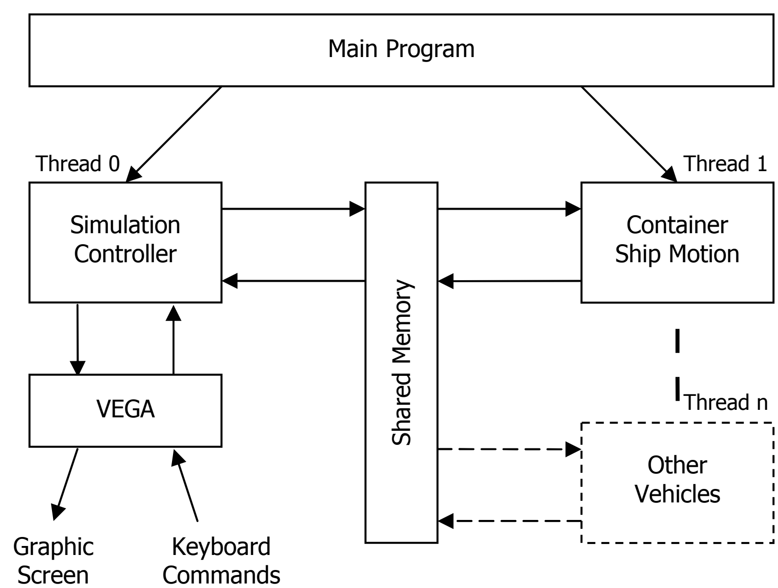



Figure 30. Simulation Program Structure.

The Main Program block creates the threads and the events needed to control the access to the shared memory area. In this thesis, only one vehicle (container ship) is implemented, but the program could accept the introduction of more vehicles with few changes.

The Simulation Controller block (*simulator_controller* thread) is responsible for controlling all the operation of the simulator. Only this thread interfaces with Vega's API. Therefore, this block controls the graphic presentation, the input of user commands and the environment management.

The Container Ship block (*container_ship_motion* thread) contains all five model implementations. The following sections describe in details these two blocks.

## B. THE SIMULATION CONTROLLER BLOCK

The content of the Simulation Controller block is depicted in Figure 31. Now, the principal functions of each one of its elements are described below.

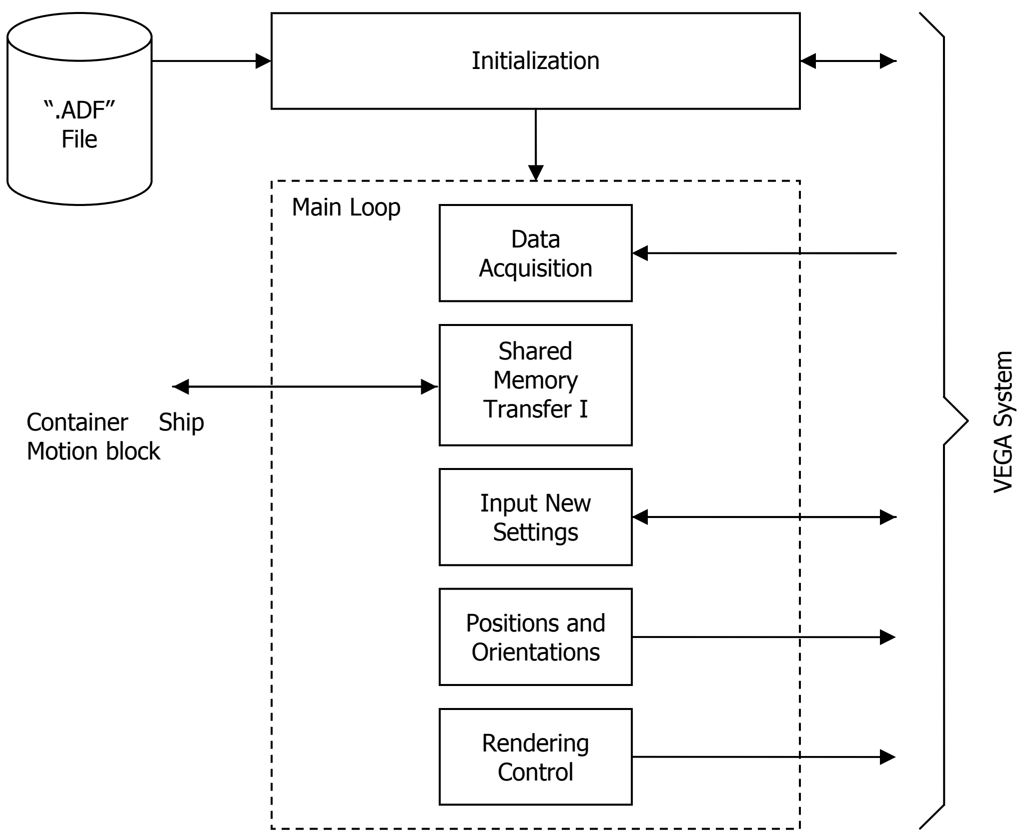



Figure 31. Simulation Controller Scheme.

### 1. Initialization

This block does the initialization of the simulation program. The initialization is divided in tasks as follows. At first, Vega and its special modules (Vega Marine and Vega Special Effects) have to be initiated. After that, the *containership_simulator.ADF* file has to be loaded to inform Vega about all the pre-made work executed in Lynx. As it was mentioned before, Lynx is a graphical interface that allows the user to construct the bulk of the application in a point and click environment. Lynx generates an Application Definition File (ADF) that is used to configure the application. Once the file is loaded, the simulation program is able to obtain all the initial settings by using the Vega's API function calls, to initiate the internal variables in order to use during the program execution.

### 2. Main Loop

This block is an infinite loop that lasts until the simulation exists. It is composed of five distinct phases:

- Data acquisition from the virtual environment in order to prepare the model's input parameters (Data Acquisition block);
- Information exchange through the critical section or shared memory. Here, the model input parameters are written and the model output results from the last iteration are read (Shared Memory Transfer I block);
- User's commands and settings (Input New Settings block);
- Position and orientation updating for the objects of the scene (Positions and Orientations block);
- Rendering permission command to Vega (Rendering Control block).

### 3. Data Acquisition

This block reads variables of interest from the current state of the environment and prepares the model inputs. There are three phases in this operation. The variables obtained in each phase are shown below:

- Wave parameters: *wave_period* (period of the fundamental wave component), *wave_heading_R* (wave heading related to the ship heading), and *alpha* (phase angle of the ship CG position related to the *wave_elev_max* point). All ocean readings are referred to the *ocean_calc*;

- Ocean current parameters: *curr_heading_R* (current heading related to the ship heading), *u_current_R* and *v_current_R* (*u* and *v* components of the current velocity in the Maneuvering Model coordinate frame);
- Wind parameters: *wind_gamma_R* (wind heading related to the ship heading), *u_current_R* and *v_current_R* (*u* and *v* components of the wind velocity in the Maneuvering Model coordinate frame).

#### 4. Shared Memory Transfer I

The following variables are written on the shared memory: *ship_status* (DRIFTING, CRUISING or SINKING), *frame_time* (integration interval), *shaft_order* and *rudder_order*, *wave_period*, *wave_heading_R*, *alpha*, *wave_elev_max*, *u_current_R* and *v_current_R*, *wind_gamma_R*, *u_current_R* and *v_current_R*, *aux_1*, *aux_2* and *aux_3* (auxiliary variables for ploting), and *trigger* (starts/finishes the recording of the log file).

The following variables are read from the shared memory: *ship_x*, *ship_y* and *ship_z* (positions), *ship_h*, *ship_r*, *ship_p* (orientations), *ship_u* and *ship_v* (velocities), *ship_n* (actual shaft speed), *ship_delta* (actual rudder angle), and *speed* (actual ship speed).

After that, the *container_ship_motion* thread is waked to provide the next set of model outputs.

#### 5. Input New Settings

This block is responsible for the input of commands and the consequent new settings in the Vega context, as required. The commands are separated in groups as follows:

- Visualization commands: change the current point of view into perspective, port side, astern, bow, bridge and above positions, and execute zoom-in and zoom-out in each of them (except bridge);
- Ship commands: change the engine and rudder orders;
- Information commands: toggle the grid, the compass, the vectors and the text on the screen, and starts/stops/clears the stopwatch;
- Environment commands: set the sea state, toggle the fog, increase and decrease the daylight, the intensity of the wind and the ocean current, and change the direction of the wind, the ocean current, and the wave-heading;

- Auxiliary commands: increase/decrease values of test variables, start/stop the recording of the log file and display variables of interest in the system command window.

### 6. Positions and Orientations

This block implements the ship status checking. The ship status could be *DRIFTING*, *CRUISING* or *SINKING.* Based on the ship status information, this block updates the position and orientation of the ship and the direction of the smoke. The position and orientation of other objects of the scene as the compass and the direction vectors are also updated here.

### 7. Rendering Control

The Rendering Control block synchronizes frame processing using the Vega's API *vgSyncFrame( )*, and starts the rendering of a new frame with *vgFrame( )*. In this block, all variables that will show up on the screen must be updated in order to be used by the post draw *drawHUD( )* function. Here, if the stopwatch is in the *COUNTING* state, it is also updated based on the frame time.

## C. THE CONTAINER SHIP BLOCK

The Container Ship block (*container_ship_motion* thread) contains all five model implementations as shown in Figure 32.

### 1. Initialization

This block does the initialization of the container ship motion models. All state variables are initialized here.

### 2. Main Loop

This block is an infinite loop that lasts until the simulation exists. It is composed of four distinct phases:

- Information exchange through the critical section or shared memory. Here, the model input parameters are read and the model output results from the last iteration are written (Shared Memory Transfer II block);
- Execution of the model code;

- Numerical integration;
- Coordinate frame conversion.

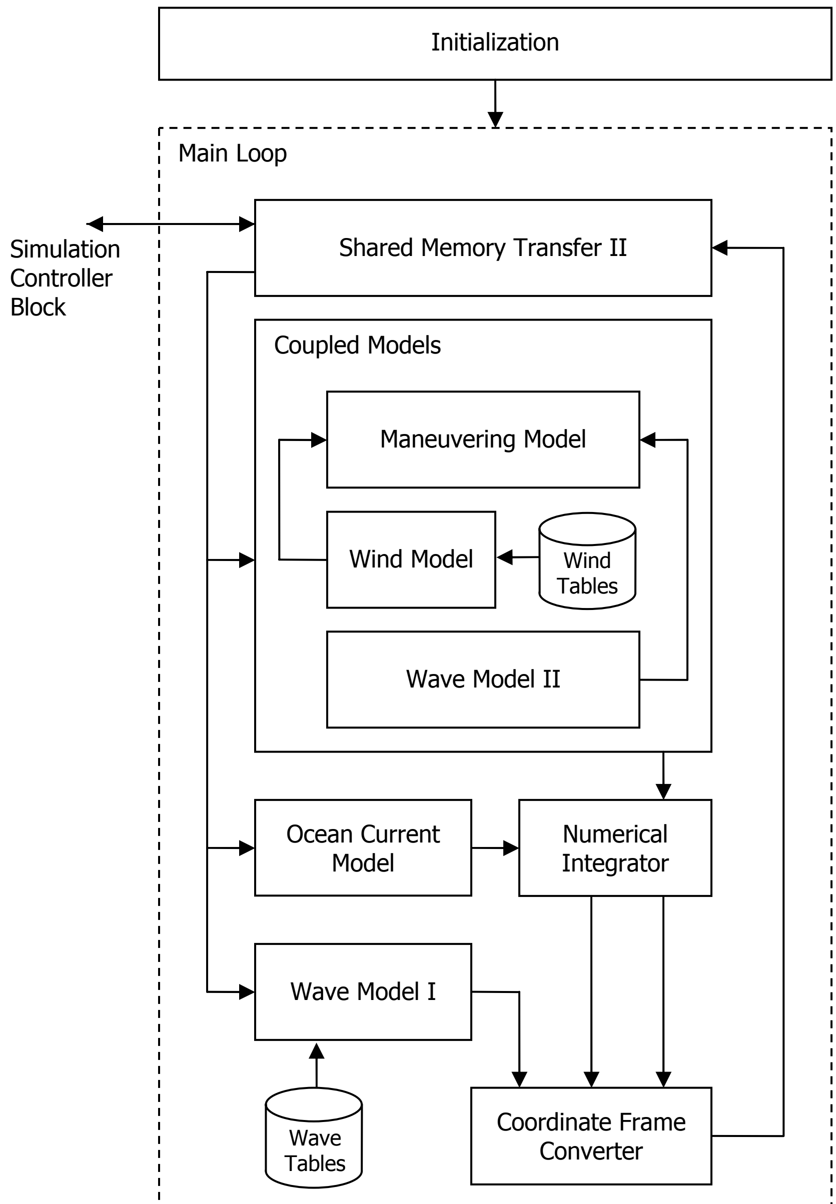



Figure 32. Container Ship Block.

### 3. Shared Memory Transfer II

The following variables are read from the shared memory: *ship_status* (DRIFTING, CRUISING or SINKING), *frame_time* (integration interval), *shaft_order* and *rudder_order*, *wave_period*, *wave_heading_R*, *alpha*, *wave_elev_max*, *u_current_R* and *v_current_R*, *wind_gamma_R*, *u_current_R* and *v_current_R*, *aux_1*, *aux_2* and *aux_3* (auxiliary variables for ploting), and *trigger* (starts/finishes the recording of the log file).

The following variables are written in the shared memory: *ship_x*, *ship_y* and *ship_z* (positions), *ship_h*, *ship_r*, *ship_p* (orientations), *ship_u* and *ship_v* (velocities), *ship_n* (actual shaft speed), *ship_delta* (actual rudder angle), and *speed* (actual ship speed).

After that, the *simulation_controller* thread is waked up to update the graphical scene with the last computed data.

#### 4. Coupled Models

This block holds three coupled models: the Maneuvering Model, the Wind Model and the Wave Model II. It is represented by the *containerShipCoupledModels (x, x_dot, input)* function. This function has the state vector *x* and the vector *input* as inputs and the state derivative vector *x_dot* as output. It is called in the drifting and cruising ship state situations. The Wind Model uses the set of tables declared in the *windModelTables.h* file. Those tables are interpolated to compute the $A_{0..6}$, $B_{0..6}$ and $C_{0..5}$ coefficients.

#### 5. Ocean Current Model

This block simply gets the surge and sway velocity components and passes them to the Numerical Integration block.

#### 6. Wave Model I

This block is represented by the *waveModel (inputWM, outputWM)* function. It is called in all ship state situations (drifting, cruising and sinking). The Wave Model I block uses the set of tables declared in the *waveModelTables.h* file.

#### 7. Numerical Integrator

This block implements the Euler's integration method to compute the next state variable values for the Coupled Models block and the Ocean Current Model block. The Numerical Integrator block uses the *frame_time* variable as integration interval.

### 8. Coordinate Frame Converter

This block implements the required coordinate transformations to adjust the model outputs to the coordinate frame used by the Vega system. Here, the model outputs are combined in order to produce the 6-DOF pack of variables that represent the position and orientation of the ship in the next graphical frame.

This chapter described how the simulation system software is organized. The next chapter shows how to operate the simulation program, describing all the commands the operator can use.

# VII. OPERATION OF THE SIMULATOR

This chapter describes the operational procedures of the ship simulation system developed in the previous chapters.

## A. GETTING STARTED

The simulation program is started by running the *shipSimulator.exe* executable file. The simulation window opens, showing the container ship drifting between two buoys. The engine is running (indicated by the ship smoke), but the clutch is not into gear. The following initial conditions are observed:

- Point of view: perspective;
- Fog: ON and daylight: maximum;
- Sea state: 1 and wave heading: 135 degrees;
- Ocean current intensity: 0 knots and direction: 0 degrees;
- Wind intensity: 0 knots and direction: 225 degrees;
- Grid: OFF, direction vectors: OFF, and compass: OFF;
- Stopwatch: reset;
- Screen text: ON.

## B. COMMANDS

The operator introduces commands using the keyboard. They are grouped in four sets: visualization, ship order, information, environment and auxiliary commands. Table 10 shows the visualization commands.

| Key | Action |
|---|---|
| *p* | Changes the point of view position to perspective, port side, astern, bow, bridge and above positions. |
| *z* | Increase zoom out |
| *Z* | Increase zoom in |

Table 10. Visualization Commands.

Table 11 shows the ship order commands.

| Key | Action |
|---|---|
| *Up arrow* | Increase engine order from Stop, $^1/_3$ Ahead, $^2/_3$ Ahead to Full Ahead |
| *Down arrow* | Decrease engine order from Full Ahead, $^2/_3$ Ahead, $^1/_3$ Ahead to Stop |
| *Right arrow* | Turn rudder to starboard up to 20 deg (1 deg steps) |
| *Left arrow* | Turn rudder to port up to −20 deg (1 deg steps) |

Table 11. Ship Order Commands.

Table 12 shows the environment commands.

| Key | Action |
|---|---|
| *f* | Toggle the fog |
| *l* | Decrease the daylight |
| *L* | Increase the daylight |
| *W* | Increase the wind direction (5 deg steps) |
| *w* | Decrease the wind direction (5 deg steps) |
| *C* | Increase the ocean current direction (5 deg steps) |
| *c* | Decrease the ocean current direction (5 deg steps) |
| *H* | Increase the wave heading angle (5 deg steps) |
| *h* | Decrease the wave heading angle (5 deg steps) |
| *E* | Increase the wind intensity up to 50 knots (5 knots steps) |
| *e* | Decrease the wind intensity down to 0 (5 knots steps) |
| *V* | Increase the current intensity up to 3 knots (0.5 knots steps) |
| *v* | Decrease the current intensity down to 0 (0.5 knots steps) |
| *1* | Set the sea state to 1 |
| *2* | Set the sea state to 2 |
| *3* | Set the sea state to 3 |
| *4* | Set the sea state to 4 |
| *5* | Set the sea state to 5 |
| *6* | Set the sea state to 6 |

Table 12. Environment Commands.

Table 13 shows the information commands.

| Key | Action |
|---|---|
| *g* | Toggle the grid |
| *a* | Toggle the direction vectors |
| *q* | Toggle the compass |
| *t* | Toggle the text on the screen |
| *s* | Command the stopwatch |

Table 13. Information Commands.

The auxiliary commands are used for testing, plotting and debugging purposes. They are used during the development phase to make easier to conduct visual tests, to create a log file for plotting and to help in debugging operations. Table 14 shows the auxiliary commands.

| Key | Action |
|---|---|
| *M/N* | Increase the value of a variable to check its visual effect |
| *m/n* | Decrease the value of a variable to check its visual effect |
| *r* | Start/stop the recording of the log file for plotting purpose |
| *d* | Display variables of interest on the system's window |

Table 14. Auxiliary Commands.

This chapter described all the commands required to operate the simulation program properly. The next chapter ends this study with the thesis conclusions and suggestions for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

# VIII. CONCLUSIONS AND RECOMMENDATIONS

The objective of this thesis was to develop a computer simulation program in which physical models were implemented in order to achieve a realistic representation of a ship in a virtual environment considering its physical features such as the engine, propeller and skull in the presence of environment conditions including waves, ocean current, wind, fog and day/night issues. This objective was reached. A simulation program was developed and it worked well. All the modeling goals defined by the initial survey were accomplished, except the weight distribution response. However, this item was of less relevance in the expert's point of view for naval tactical simulators. The program is a good starting point for further implementations

Complete marine models are difficult to find in the literature probably because their construction depends on huge experimental efforts to define all the hydrodynamic derivative coefficients. However, this study showed that the integration of multiple models from different sources is feasible and produces good results, meeting the application requirements.

The use of the interpolation technique to take advantage of complex models worked well. The linear interpolation of pre-assembled numerical tables produces a very low computational load when running in real-time and the results were very acceptable.

The Vega system showed to be very useful to build complex virtual environment applications in a short time. Some special effects from Vega Marine module are very good and the ocean simulation is fine. The bow wave effect, however, does not work well especially in high seas conditions. The number of API functions is large and they are easy to use. The built-in manual is very useful too.

The use of dynamic models with relative low complexity produced sufficiently good results, saving processing time for other tasks in complex simulation systems.

As a recommendation for future work, a natural way of continuing this thesis work would be the integration with sensor models, such as radar models, that consider the sea state and the weather conditions as inputs to impose some constrains in their usage.

Weapons could be included in the experiment as well. Naval tactical simulators that include all these features should be very useful to make the officer's training more realistic and immersive.

# APPENDIX A. SURVEY

If you were going to use a Simulator for Naval Tactical Training in a 3D Virtual Environment with the following features:

- 3D graphical scenery of operations where you are able to visually track the maneuvers of the participant platforms (ships, helicopters, airplanes, submarines) from several points of view.
- All platforms are equipped with full armament and detection equipments.
- The system allows engagement between platforms.

**Question:** What would you consider **IMPORTANT MODELING PARAMETERS** in terms of physical characteristics of the **SHIPS** and the **ENVIRONMENT** that could make the **TRAINING** more realistic?

**Note 1:** Please, rank the options below starting from the most to the least relevant in your opinion, 1 being most relevant. More than one option can have the same level of relevance.

**Note 2:** In your evaluation, consider **PRIORITIZING** the enhancement of **FUNCTIONAL** issues of the Simulator over esthetic issues.

____The ship behavior in terms of acceleration/deceleration rates

____The ship behavior in terms of turning rate

____The ship behavior in terms of weight distribution (tilt/trim)

____The wake of the ship

____The influence of sea state (ship "rocking")

____The influence of currents in the ship behavior

____The influence of the wind in the ship behavior

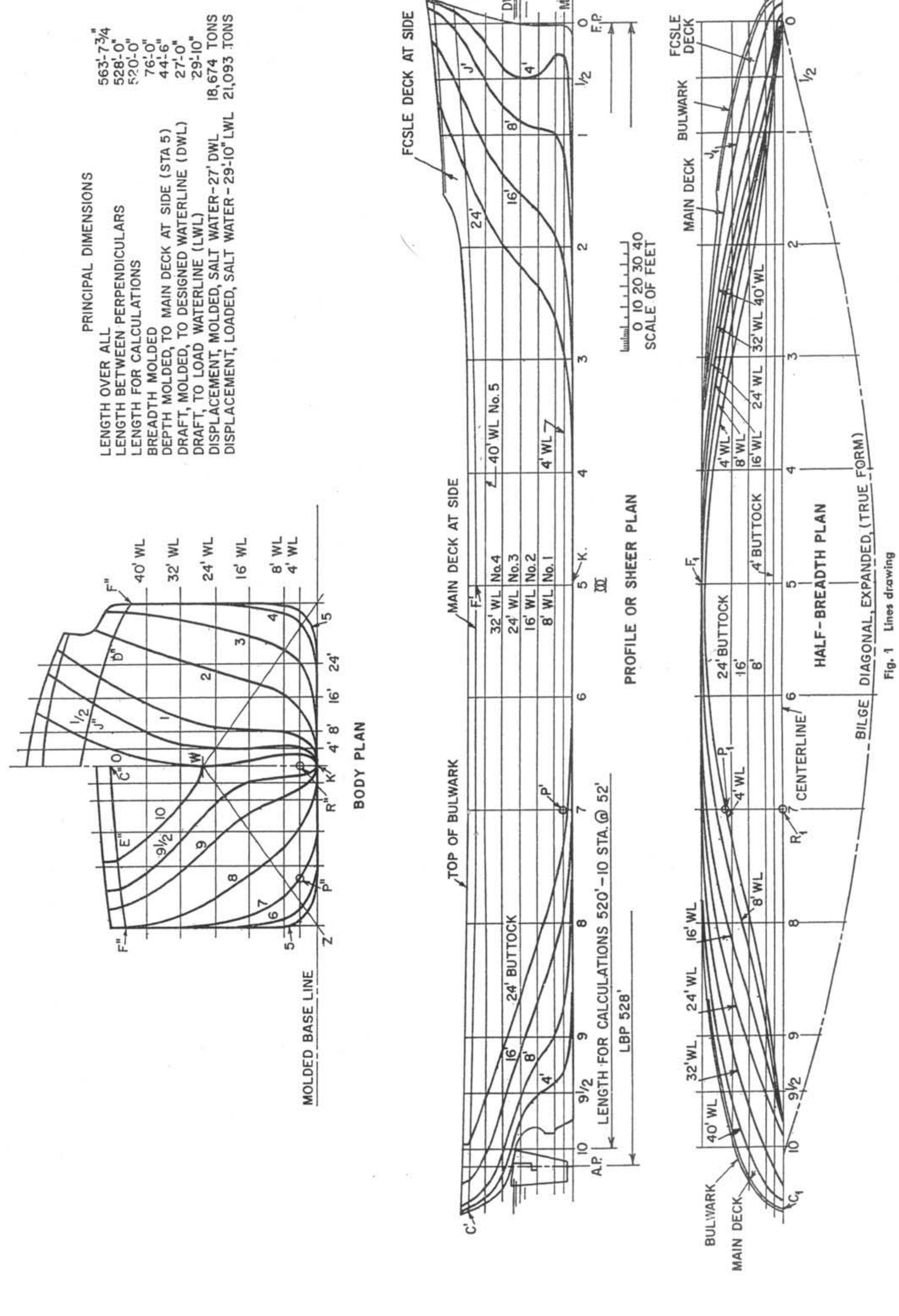
____Visibility issues (day/night, fog)

____Other: ____________________________________________

**Your area of activity:** __________________________________

THIS PAGE INTENTIONALLY LEFT BLANK

# **APPENDIX B. CONTAINER SHIP BODY PLAN** [AFTER REF. 10]



THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX C. SHIPMO.IN INPUT FILE

The *SHIPMO.IN* file showed below is the input file to generate the numerical tables by running the *SHIPMO.BM* program.

```
Container ship model
    0    0    0    0    1    0    1    1    1    0    1    0    0   13    0
  520.0000    1.9905   32.1740       1.26E-05            0.0    0.0000
   33.0000  -26.0000    1.0000
    7  260.0000    0.0000    0
    0.0000  -29.0000

    1.1200  -28.5000
    2.5300  -25.0000
    2.4900  -21.0000
    1.1200  -13.0000
    0.0100   -2.0000
    0.5600    0.0000
    7  234.0000    0.0000    0
    0.0000  -29.0000
    1.9600  -28.5000
    4.2100  -25.0000
    4.7700  -21.0000
    4.2100  -13.0000
    4.4900   -2.0000
    4.8000    0.0000
    7  208.0000    0.0000    0
    0.0000  -29.0000
    3.3700  -28.5000
    6.7400  -25.0000
    8.1400  -21.0000
    8.4200  -13.0000
    9.8200   -2.0000
   10.1600    0.0000
    7  156.0000    0.0000    0
    0.0000  -29.0000
    6.1800  -28.5000
   13.1900  -25.0000
   16.8400  -21.0000
   19.6500  -13.0000
   23.0200   -2.0000
   24.0000    0.0000
    7  104.0000    0.0000    0
    0.0000  -29.0000
   12.9100  -28.5000
   24.0000  -25.0000
   27.2300  -21.0000
   30.3200  -13.0000
   33.1200   -2.0000
   33.3200    0.0000
    7   52.0000    0.0000    0
    0.0000  -29.0000
   24.9800  -28.5000
   33.1200  -25.0000
   35.3700  -21.0000
   36.7700  -13.0000
   37.8900   -2.0000
   37.8900    0.0000
    7    0.0000    0.0000    0
    0.0000  -29.0000
   29.4700  -28.5000
   35.6500  -25.0000
   37.0500  -21.0000
```

```
37.8900  -13.0000
37.8900   -2.0000
37.8900    0.0000
 7  -52.0000    0.0000    0

 0.0000  -29.0000
24.9800  -28.5000
33.1200  -25.0000
35.3700  -21.0000
36.7700  -13.0000
37.8900   -2.0000
37.8900    0.0000
 7 -104.0000    0.0000    0
 0.0000  -29.0000
18.9200  -28.5000
26.5400  -25.0000
30.4900  -21.0000
35.0100  -13.0000
37.2700   -2.0000
37.5500    0.0000
 7 -156.0000    0.0000    0
 0.0000  -29.0000
 6.7800  -28.5000
14.4000  -25.0000
19.2000  -21.0000
25.4100  -13.0000
32.4700   -2.0000
33.6000    0.0000
 7 -208.0000    0.0000    0
 0.0000  -29.0000
 2.2500  -28.5000
 5.3600  -25.0000
 7.3400  -21.0000
11.0100  -13.0000
20.0500   -2.0000
22.0200    0.0000
 7 -234.0000    0.0000    0
 0.0000  -29.0000
 1.1300  -28.5000
 2.2600  -25.0000
 2.8200  -21.0000
 3.6700  -13.0000
11.8600   -2.0000
13.8400    0.0000
 2 -260.0000    0.0000    0
 0.0000   -2.0000
 4.8000    0.0000
   -2.0    0.0000
 9999.0       0.0       0.0
 1.0000   10.0000 1500.0000   50.0000   00.0000   60.0000   10.0000
 0.0000  180.0000   15.0000
```

# APPENDIX D. SHIPMO.OUT OUTPUT FILE

The *SHIPMO.OUT* file is the output file of the *SHIPMO.BM* program. Since it is a very large file, only the initial information and the two first sets of tables (of a total of 91 tables) are shown below to provide an idea of its format.

```
1Container ship model
0OPTION CONTROL TAGS -  A  B  C  D  E  F  G  H  I  J  K  L  M  N NS NPAC
                        0  0  0  0  1  0  1  1  1  0  1  0 13  0 13  0
0***BASIC INPUT DATA***

          *******   INFINITE DEPTH  *******
0 LENGTH=   520.000
  DENSITY=    1.9905  GAMMA=   64.0423  GNU=   0.126000E-04  GRAVITY=   32.1740
0BILGE KEEL DATA:  XFOR=   33.0000  XAFT=  -26.0000  LENGTH=   59.0000  WIDTH=    1.0000
0STA NO   XAXIS     1/2BEAM    DRAFT    AREA     AREA COEF   ZBAR    BILGE R  AREA COEF2
     1   260.0000    0.5600   29.0000   75.8650    2.3357  -18.9654    0.0000    2.3357
     2   234.0000    4.8000   29.0000  244.6150    0.8786  -13.8322    0.0000    0.8786
     3   208.0000   10.1600   29.0000  469.6700    0.7970  -13.0276    0.0000    0.7970
     4   156.0000   24.0000   29.0000 1046.3350    0.7517  -12.5382    0.0000    0.7517
     5   104.0000   33.3200   29.0000 1631.6801    0.8443  -13.1316    0.0000    0.8443
     6    52.0000   37.8900   29.0000 2039.7400    0.9282  -13.8060    0.0000    0.9282
     7     0.0000   37.8900   29.0000 2118.1150    0.9638  -14.0585    0.0000    0.9638
     8   -52.0000   37.8900   29.0000 2039.7400    0.9282  -13.8060    0.0000    0.9282
     9  -104.0000   37.5500   29.0000 1865.4099    0.8565  -13.2276    0.0000    0.8565
    10  -156.0000   33.6000   29.0000 1337.6200    0.6864  -11.8926    0.0000    0.6864
    11  -208.0000   22.0200   29.0000  651.1600    0.5098  -10.4750    0.0000    0.5098
    12  -234.0000   13.8400   29.0000  306.9000    0.3823   -9.2436    0.0000    0.3823
    13  -260.0000    4.8000    2.0000    9.6000    0.5000   -0.6667    0.0000    0.5000
0THE FOLLOWING STATIONS AND OFFSETS ARE USED FOR TWO-DIMENSIONAL CALCULATIONS
  STA NO=  1  XAXIS=  260.0000  NT=  7  ILID=  0
    0.0000  -29.0000
    1.1200  -28.5000
    2.5300  -25.0000
    2.4900  -21.0000
    1.1200  -13.0000
    0.0100   -2.0000
    0.5600    0.0000
  STA NO=  2  XAXIS=  234.0000  NT=  7  ILID=  0
    0.0000  -29.0000
    1.9600  -28.5000
    4.2100  -25.0000
    4.7700  -21.0000
    4.2100  -13.0000
    4.4900   -2.0000
    4.8000    0.0000
  STA NO=  3  XAXIS=  208.0000  NT=  7  ILID=  0
    0.0000  -29.0000
    3.3700  -28.5000
    6.7400  -25.0000
    8.1400  -21.0000
    8.4200  -13.0000
    9.8200   -2.0000
   10.1600    0.0000
  STA NO=  4  XAXIS=  156.0000  NT=  7  ILID=  0
    0.0000  -29.0000
    6.1800  -28.5000
   13.1900  -25.0000
   16.8400  -21.0000
   19.6500  -13.0000
   23.0200   -2.0000
   24.0000    0.0000
```

```
  STA NO=  5  XAXIS=  104.0000  NT=  7  ILID=  0
    0.0000  -29.0000
   12.9100  -28.5000
   24.0000  -25.0000
   27.2300  -21.0000
   30.3200  -13.0000
   33.1200   -2.0000

   33.3200    0.0000
  STA NO=  6  XAXIS=   52.0000  NT=  7  ILID=  0
    0.0000  -29.0000
   24.9800  -28.5000
   33.1200  -25.0000
   35.3700  -21.0000
   36.7700  -13.0000
   37.8900   -2.0000
   37.8900    0.0000
  STA NO=  7  XAXIS=    0.0000  NT=  7  ILID=  0
    0.0000  -29.0000
   29.4700  -28.5000
   35.6500  -25.0000
   37.0500  -21.0000
   37.8900  -13.0000
   37.8900   -2.0000
   37.8900    0.0000
  STA NO=  8  XAXIS=  -52.0000  NT=  7  ILID=  0
    0.0000  -29.0000
   24.9800  -28.5000
   33.1200  -25.0000
   35.3700  -21.0000
   36.7700  -13.0000
   37.8900   -2.0000
   37.8900    0.0000
  STA NO=  9  XAXIS= -104.0000  NT=  7  ILID=  0
    0.0000  -29.0000
   18.9200  -28.5000
   26.5400  -25.0000
   30.4900  -21.0000
   35.0100  -13.0000
   37.2700   -2.0000
   37.5500    0.0000
  STA NO= 10  XAXIS= -156.0000  NT=  7  ILID=  0
    0.0000  -29.0000
    6.7800  -28.5000
   14.4000  -25.0000
   19.2000  -21.0000
   25.4100  -13.0000
   32.4700   -2.0000
   33.6000    0.0000
  STA NO= 11  XAXIS= -208.0000  NT=  7  ILID=  0
    0.0000  -29.0000
    2.2500  -28.5000
    5.3600  -25.0000
    7.3400  -21.0000
   11.0100  -13.0000
   20.0500   -2.0000
   22.0200    0.0000
  STA NO= 12  XAXIS= -234.0000  NT=  7  ILID=  0
    0.0000  -29.0000
    1.1300  -28.5000
    2.2600  -25.0000
    2.8200  -21.0000
    3.6700  -13.0000
   11.8600   -2.0000
   13.8400    0.0000
  STA NO= 13  XAXIS= -260.0000  NT=  2  ILID=  0
    0.0000   -2.0000
    4.8000    0.0000
0COMPARISON OF INPUT AND COMPUTED DATA
  INPUT       DISPL=     19648.7637  XCG=   -7.5667 RELATIVE TO ORIGIN AT MIDSHIP
  HYDROSTSTIC DISPL=     19648.7637  XCG=   -7.5667 RELATIVE TO ORIGIN AT MIDSHIP
```

```
0BLOCK COEFFICIENT=    0.6014
0STABILITY PARAMETERS
          LCF =  -18.1800  ZCG =   -2.0000  ZCB =  -13.1960
          WATERPLANE AREA =     28801.2402
          BML =  598.0724  GML =  586.8764
          BMT =   15.5936  GMT =    4.3976
0RADII OF GYRATION
          KYY =  130.0000 KXX =   26.5230 PRODUCT OF INERTIA(I46) =   0.207022E+08
1Container ship model


 ***CONDITIONAL INPUT DATA***

  WAVE AMPLITUDE=    1.0000
  INITIAL WAVELENGTH=   10.0000 FINAL WAVELENGTH= 1500.0000 DELTA WAVELENGTH=   50.0000
  INITIAL VEL=    0.0000 FINAL VEL=   60.0000  DELTA VEL=   10.0000
  INITIAL WAVE HEADING ANGLE=    0.0000 FINAL WAVE HEADING ANGLE=  180.0000
                              DELTA WAVE HEADING ANGLE=   15.0000



 ***UNITS OF OUTPUT***
          NONDIMENSIONAL OUTPUT
            OUTPUT IS DIVIDED BY:
               LINEAR MOTIONS - WAVE AMPLITUDE (WA)
               ROTATIONAL MOTIONS - WAVE SLOPE (WA*WAVEN)
               VELOCITIES - WA*SQRT(GRAV/BPL)
               ACCELERATIONS - WA*GRAV/BPL
               SHEAR - WA*GAMMA*BEAM*BPL
               BENDING MOMENTS - WA*GAMMA*BEAM*BPL**2
1Container ship model
0 SPEED =  0.0000      WAVE ANGLE =    0.00 DEG.
+                                                    VERTICAL PLANE RESPONSES
+                                                        (NON-DIMENSIONAL)
```

| WAVE FREQUENCIES | ENCOUNTER FREQUENCIES | WAVE LENGTH | WAVE/SHIP LENGTH | SURGE AMPL. | SURGE PHASE | HEAVE AMPL. | HEAVE PHASE | PITCH AMPL. | PITCH PHASE |
|---|---|---|---|---|---|---|---|---|---|
| 4.49617 | 4.49617 | 10.000 | 0.0192 | 0.0000 | -173.2 | 0.0000 | -148.8 | 0.0000 | -154.2 |
| 1.83555 | 1.83555 | 60.000 | 0.1154 | 0.0008 | -113.5 | 0.0003 | -97.8 | 0.0001 | -125.5 |
| 1.35565 | 1.35565 | 110.000 | 0.2115 | 0.0048 | -100.1 | 0.0036 | -133.3 | 0.0019 | -121.0 |
| 1.12404 | 1.12404 | 160.000 | 0.3077 | 0.0042 | -11.5 | 0.0265 | -37.7 | 0.0152 | -15.2 |
| 0.98114 | 0.98114 | 210.000 | 0.4038 | 0.0164 | -90.5 | 0.0536 | -112.6 | 0.0610 | -135.7 |
| 0.88177 | 0.88177 | 260.000 | 0.5000 | 0.0152 | 19.6 | 0.1493 | 136.0 | 0.1379 | -150.8 |
| 0.80754 | 0.80754 | 310.000 | 0.5962 | 0.0217 | 32.4 | 0.2701 | 152.4 | 0.0923 | 174.1 |
| 0.74936 | 0.74936 | 360.000 | 0.6923 | 0.0339 | -74.7 | 0.1904 | 166.9 | 0.1795 | 112.7 |
| 0.70218 | 0.70218 | 410.000 | 0.7885 | 0.1055 | -88.6 | 0.0507 | -175.2 | 0.3214 | 102.0 |
| 0.66292 | 0.66292 | 460.000 | 0.8846 | 0.1852 | -90.0 | 0.0912 | -15.1 | 0.4431 | 98.9 |
| 0.62959 | 0.62959 | 510.000 | 0.9808 | 0.2619 | -90.2 | 0.2137 | -8.4 | 0.5413 | 97.5 |
| 0.60083 | 0.60083 | 560.000 | 1.0769 | 0.3317 | -90.1 | 0.3178 | -5.9 | 0.6202 | 96.7 |
| 0.57568 | 0.57568 | 610.000 | 1.1731 | 0.3935 | -90.1 | 0.4053 | -4.6 | 0.6833 | 96.1 |
| 0.55344 | 0.55344 | 660.000 | 1.2692 | 0.4477 | -90.0 | 0.4787 | -3.7 | 0.7343 | 95.6 |
| 0.53360 | 0.53360 | 710.000 | 1.3654 | 0.4950 | -90.0 | 0.5403 | -3.1 | 0.7760 | 95.2 |
| 0.51575 | 0.51575 | 760.000 | 1.4615 | 0.5362 | -90.0 | 0.5922 | -2.6 | 0.8102 | 94.9 |
| 0.49957 | 0.49957 | 810.000 | 1.5577 | 0.5721 | -90.0 | 0.6364 | -2.3 | 0.8384 | 94.6 |
| 0.48483 | 0.48483 | 860.000 | 1.6538 | 0.6035 | -90.0 | 0.6740 | -2.0 | 0.8620 | 94.4 |
| 0.47133 | 0.47133 | 910.000 | 1.7500 | 0.6312 | -90.1 | 0.7063 | -1.8 | 0.8819 | 94.1 |
| 0.45889 | 0.45889 | 960.000 | 1.8462 | 0.6556 | -90.1 | 0.7343 | -1.6 | 0.8987 | 93.9 |
| 0.44739 | 0.44739 | 1010.000 | 1.9423 | 0.6771 | -90.1 | 0.7585 | -1.4 | 0.9130 | 93.8 |
| 0.43671 | 0.43671 | 1060.000 | 2.0385 | 0.6963 | -90.2 | 0.7797 | -1.3 | 0.9253 | 93.6 |
| 0.42676 | 0.42676 | 1110.000 | 2.1346 | 0.7134 | -90.3 | 0.7983 | -1.2 | 0.9358 | 93.4 |
| 0.41746 | 0.41746 | 1160.000 | 2.2308 | 0.7287 | -90.3 | 0.8147 | -1.1 | 0.9450 | 93.3 |
| 0.40874 | 0.40874 | 1210.000 | 2.3269 | 0.7425 | -90.4 | 0.8292 | -1.0 | 0.9530 | 93.1 |
| 0.40055 | 0.40055 | 1260.000 | 2.4231 | 0.7549 | -90.5 | 0.8421 | -0.9 | 0.9599 | 93.0 |
| 0.39283 | 0.39283 | 1310.000 | 2.5192 | 0.7662 | -90.5 | 0.8536 | -0.9 | 0.9661 | 92.9 |
| 0.38554 | 0.38554 | 1360.000 | 2.6154 | 0.7764 | -90.6 | 0.8640 | -0.8 | 0.9715 | 92.8 |
| 0.37865 | 0.37865 | 1410.000 | 2.7115 | 0.7857 | -90.6 | 0.8733 | -0.8 | 0.9763 | 92.7 |
| 0.37211 | 0.37211 | 1460.000 | 2.8077 | 0.7942 | -90.7 | 0.8817 | -0.7 | 0.9805 | 92.6 |

1Container ship model

0 SPEED = 0.0000 WAVE ANGLE = 15.00 DEG.

+ VERTICAL PLANE RESPONSES

+ (NON-DIMENSIONAL)

| WAVE FREQUENCIES | ENCOUNTER FREQUENCIES | WAVE LENGTH | WAVE/SHIP LENGTH | SURGE AMPL. | SURGE PHASE | HEAVE AMPL. | HEAVE PHASE | PITCH AMPL. | PITCH PHASE |
|---|---|---|---|---|---|---|---|---|---|
| 4.49617 | 4.49617 | 10.000 | 0.0192 | 0.0000 | -125.6 | 0.0000 | -110.8 | 0.0000 | -107.4 |
| 1.83555 | 1.83555 | 60.000 | 0.1154 | 0.0004 | -176.0 | 0.0004 | 146.0 | 0.0002 | -172.3 |
| 1.35565 | 1.35565 | 110.000 | 0.2115 | 0.0037 | -118.6 | 0.0045 | -168.1 | 0.0020 | -153.9 |
| 1.12404 | 1.12404 | 160.000 | 0.3077 | 0.0069 | -48.8 | 0.0305 | -53.5 | 0.0156 | -30.8 |
| 0.98114 | 0.98114 | 210.000 | 0.4038 | 0.0150 | -93.8 | 0.0484 | -133.0 | 0.0691 | -144.1 |
| 0.88177 | 0.88177 | 260.000 | 0.5000 | 0.0147 | 23.4 | 0.1784 | 135.4 | 0.1312 | -157.0 |
| 0.80754 | 0.80754 | 310.000 | 0.5962 | 0.0156 | 17.5 | 0.2689 | 153.1 | 0.0941 | 153.2 |
| 0.74936 | 0.74936 | 360.000 | 0.6923 | 0.0465 | -83.9 | 0.1638 | 168.1 | 0.2117 | 107.2 |
| 0.70218 | 0.70218 | 410.000 | 0.7885 | 0.1213 | -90.4 | 0.0179 | -140.1 | 0.3516 | 100.0 |
| 0.66292 | 0.66292 | 460.000 | 0.8846 | 0.2004 | -90.8 | 0.1313 | -12.5 | 0.4668 | 97.9 |
| 0.62959 | 0.62959 | 510.000 | 0.9808 | 0.2747 | -90.7 | 0.2531 | -7.7 | 0.5580 | 96.9 |
| 0.60083 | 0.60083 | 560.000 | 1.0769 | 0.3415 | -90.5 | 0.3548 | -5.6 | 0.6304 | 96.2 |
| 0.57568 | 0.57568 | 610.000 | 1.1731 | 0.4001 | -90.3 | 0.4395 | -4.3 | 0.6878 | 95.7 |
| 0.55344 | 0.55344 | 660.000 | 1.2692 | 0.4510 | -90.2 | 0.5098 | -3.5 | 0.7339 | 95.3 |
| 0.53360 | 0.53360 | 710.000 | 1.3654 | 0.4954 | -90.2 | 0.5686 | -2.9 | 0.7715 | 94.9 |
| 0.51575 | 0.51575 | 760.000 | 1.4615 | 0.5338 | -90.1 | 0.6179 | -2.5 | 0.8021 | 94.6 |
| 0.49957 | 0.49957 | 810.000 | 1.5577 | 0.5673 | -90.1 | 0.6597 | -2.1 | 0.8273 | 94.4 |
| 0.48483 | 0.48483 | 860.000 | 1.6538 | 0.5965 | -90.2 | 0.6952 | -1.9 | 0.8483 | 94.1 |
| 0.47133 | 0.47133 | 910.000 | 1.7500 | 0.6221 | -90.2 | 0.7257 | -1.7 | 0.8659 | 93.9 |
| 0.45889 | 0.45889 | 960.000 | 1.8462 | 0.6447 | -90.2 | 0.7520 | -1.5 | 0.8808 | 93.8 |
| 0.44739 | 0.44739 | 1010.000 | 1.9423 | 0.6647 | -90.3 | 0.7747 | -1.4 | 0.8935 | 93.6 |
| 0.43671 | 0.43671 | 1060.000 | 2.0385 | 0.6824 | -90.3 | 0.7946 | -1.2 | 0.9043 | 93.4 |
| 0.42676 | 0.42676 | 1110.000 | 2.1346 | 0.6982 | -90.4 | 0.8120 | -1.1 | 0.9136 | 93.3 |
| 0.41746 | 0.41746 | 1160.000 | 2.2308 | 0.7123 | -90.4 | 0.8273 | -1.0 | 0.9217 | 93.1 |
| 0.40874 | 0.40874 | 1210.000 | 2.3269 | 0.7251 | -90.5 | 0.8409 | -1.0 | 0.9287 | 93.0 |
| 0.40055 | 0.40055 | 1260.000 | 2.4231 | 0.7366 | -90.6 | 0.8530 | -0.9 | 0.9348 | 92.9 |
| 0.39283 | 0.39283 | 1310.000 | 2.5192 | 0.7469 | -90.6 | 0.8637 | -0.8 | 0.9401 | 92.8 |
| 0.38554 | 0.38554 | 1360.000 | 2.6154 | 0.7564 | -90.7 | 0.8734 | -0.8 | 0.9449 | 92.7 |
| 0.37865 | 0.37865 | 1410.000 | 2.7115 | 0.7650 | -90.8 | 0.8821 | -0.7 | 0.9490 | 92.6 |
| 0.37211 | 0.37211 | 1460.000 | 2.8077 | 0.7729 | -90.8 | 0.8899 | -0.7 | 0.9527 | 92.5 |

1Container ship model

0 SPEED = 0.0000 WAVE ANGLE = 15.00 DEG.

+ LATERAL PLANE RESPONSES

+ (NON-DIMENSIONAL)

| WAVE FREQUENCIES | ENCOUNTER FREQUENCIES | WAVE LENGTH | WAVE/SHIP LENGTH | SWAY AMPL. | SWAY PHASE | ROLL AMPL. | ROLL PHASE | YAW AMPL. | YAW PHASE |
|---|---|---|---|---|---|---|---|---|---|
| 4.49617 | 4.49617 | 10.000 | 0.0192 | 0.0000 | -122.7 | 0.0000 | 177.0 | 0.0000 | -0.3 |
| 1.83555 | 1.83555 | 60.000 | 0.1154 | 0.0010 | -149.2 | 0.0002 | -123.7 | 0.0001 | -60.0 |
| 1.35565 | 1.35565 | 110.000 | 0.2115 | 0.0061 | -138.7 | 0.0023 | -125.1 | 0.0002 | 174.1 |
| 1.12404 | 1.12404 | 160.000 | 0.3077 | 0.0039 | 78.2 | 0.0041 | 68.3 | 0.0034 | 146.3 |
| 0.98114 | 0.98114 | 210.000 | 0.4038 | 0.0171 | -118.4 | 0.0257 | -97.2 | 0.0009 | -52.4 |
| 0.88177 | 0.88177 | 260.000 | 0.5000 | 0.0033 | 87.7 | 0.0464 | -88.1 | 0.0106 | -15.8 |
| 0.80754 | 0.80754 | 310.000 | 0.5962 | 0.0188 | 81.0 | 0.0364 | -76.4 | 0.0064 | 2.3 |
| 0.74936 | 0.74936 | 360.000 | 0.6923 | 0.0154 | 85.2 | 0.0089 | -28.8 | 0.0098 | 153.4 |
| 0.70218 | 0.70218 | 410.000 | 0.7885 | 0.0011 | -154.2 | 0.0350 | 93.0 | 0.0296 | 165.4 |
| 0.66292 | 0.66292 | 460.000 | 0.8846 | 0.0221 | -95.1 | 0.0779 | 104.0 | 0.0496 | 169.9 |
| 0.62959 | 0.62959 | 510.000 | 0.9808 | 0.0443 | -93.2 | 0.1242 | 108.8 | 0.0681 | 172.4 |
| 0.60083 | 0.60083 | 560.000 | 1.0769 | 0.0653 | -91.7 | 0.1724 | 112.0 | 0.0842 | 174.2 |
| 0.57568 | 0.57568 | 610.000 | 1.1731 | 0.0845 | -90.9 | 0.2255 | 114.2 | 0.0982 | 175.6 |
| 0.55344 | 0.55344 | 660.000 | 1.2692 | 0.1017 | -90.4 | 0.2855 | 115.7 | 0.1101 | 176.7 |
| 0.53360 | 0.53360 | 710.000 | 1.3654 | 0.1168 | -89.9 | 0.3553 | 116.7 | 0.1201 | 177.6 |
| 0.51575 | 0.51575 | 760.000 | 1.4615 | 0.1303 | -89.5 | 0.4410 | 117.3 | 0.1287 | 178.5 |
| 0.49957 | 0.49957 | 810.000 | 1.5577 | 0.1425 | -89.1 | 0.5507 | 117.8 | 0.1359 | 179.4 |
| 0.48483 | 0.48483 | 860.000 | 1.6538 | 0.1537 | -88.8 | 0.6987 | 118.2 | 0.1419 | -179.7 |
| 0.47133 | 0.47133 | 910.000 | 1.7500 | 0.1642 | -88.3 | 0.9131 | 118.5 | 0.1466 | -178.5 |
| 0.45889 | 0.45889 | 960.000 | 1.8462 | 0.1749 | -87.6 | 1.2548 | 119.2 | 0.1499 | -176.8 |
| 0.44739 | 0.44739 | 1010.000 | 1.9423 | 0.1870 | -86.5 | 1.8881 | 120.7 | 0.1509 | -173.9 |
| 0.43671 | 0.43671 | 1060.000 | 2.0385 | 0.2048 | -83.2 | 3.4454 | 126.6 | 0.1442 | -167.2 |
| 0.42676 | 0.42676 | 1110.000 | 2.1346 | 0.2053 | -64.1 | 8.6364 | 172.0 | 0.0601 | -164.6 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.41746 | 0.41746 | 1160.000 | 2.2308 | 0.1216 | -92.2 | 6.3591 | -92.0 | 0.1876 | 152.1 |
| 0.40874 | 0.40874 | 1210.000 | 2.3269 | 0.1634 | -94.2 | 3.0219 | -72.8 | 0.1903 | 167.0 |
| 0.40055 | 0.40055 | 1260.000 | 2.4231 | 0.1775 | -92.8 | 1.9803 | -69.7 | 0.1882 | 171.2 |
| 0.39283 | 0.39283 | 1310.000 | 2.5192 | 0.1857 | -92.1 | 1.4974 | -68.9 | 0.1880 | 173.2 |
| 0.38554 | 0.38554 | 1360.000 | 2.6154 | 0.1915 | -91.6 | 1.2200 | -68.8 | 0.1886 | 174.3 |
| 0.37865 | 0.37865 | 1410.000 | 2.7115 | 0.1962 | -91.3 | 1.0401 | -68.9 | 0.1896 | 175.1 |
| 0.37211 | 0.37211 | 1460.000 | 2.8077 | 0.2001 | -91.1 | 0.9139 | -69.2 | 0.1906 | 175.7 |

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


1. Giacoppo, A. S., *Development Methods: User Needs Assessment & Task Analyses*, Department of Psychology, Catholic University, October 2001.

2. Sandeep, S. and Zyda, M. J., *Networked Virtual Environments*, Addison Wesley, 1999.

3. Fossen, T. I., *Guidance and Control of Ocean Vehicles*, John Wiley & Sons, 1994.

4. *Vega Programmer's Guide Version 3.3 for Windows NT*, MultiGen-Paradigm, Inc., 1999.

5. *Marine User's Guide Version 3.3*, MultiGen-Paradigm, Inc., 1999.

6. Beck, R. F. and Troesch, A. W., *Documentation and User's Manual for the Computer Program SHIPMO.BM.*, 2775 Tessmer Rd., Ann Arbor, Michigan, 48103, 1989.

7. Salvesen, N., Tuck, E. O. and Faltinsen, O., "Ship Motions and Sea Loads", *The Society of Naval Architects and Marine Engineers*, Vol. 78, pp. 250-287, 1970.

8. Beck, R. F., Cao, Y., Scorpio, S. M. and Schultz, W. W., "Nonlinear Ship Motion Computations Using the Singularized Method", *Proc. 20$^{th}$ Symposium of Naval Hydrodynamics*, Santa Barbara, CA, pp. 227-246, 1994.

9. Naval Ship R&D Center, Hydrodynamics Laboratory, Technical Note 69, *On the Oscillation of Cylinders In or Below the Free Surface of Deep Fluids*, by Frank, W., 1967.

10. Comstock, J. P., "Principles of Naval Architecture", *The Society of Naval Architects and Marine Engineers*, Vol. 134, p. 3, 1977.

11. Faltisen, O. M., *Sea Loads on Ships and Offshore Structures*, Cambridge University Press, 1990.

12. Son, K. H. and Nomoto, K., "On the Coupled Motion of Steering and Rolling of a High Speed Container Ship", *Naval Architect of Ocean Engineering,* Vol. 20, pp. 73-83, 1982.

13. Slotine, J. J. E. and Li, W., *Applied Nonlinear Control*, Prentice Hall, 1991.

14. Abkowitz, M. A., *Stability and Motion Control of Ocean Vehicles*, M.I.T. Press, 2$^{nd}$ Ed., 1972.

15. Van Amerongen, J., *Adaptive Steering of Ships – A Model Reference Approach to Improved Maneuvering and Economical Course Keeping*, Ph.D. Dissertation, Delft University of Technology, The Netherlands, 1982.

16. Fossen, T. I., *Nonlinear Modeling and Control of Underwater Vehicles*, Dr. Ing. thesis, Dept. of Engineering Cybernetics, The Norwegian Institute of Technology, Trondheim, 1991.

17. Balchen, J. G., Jenssen, N. A. and Saelid, S., *Dynamic Positioning Using Kalman Filtering and Optimal Control Theory,* IFAC/IFIP Symposium on Automation in Offshore Oil Field Operation, Holland, Amsterdam, pp. 183-186, 1976.

18. Isherwood, R. M., "Wind Resistance of Merchant Ships", *RINA Trans.*, Vol. 115, pp. 327-338, 1972.

19. Kempf, G., "Measurements of the Propulsive and Structural Characteristics of Ships", *Transactions of SNAME*, Vol. 40, pp. 42-57, 1932.

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia

2. Dudley Knox Library
Naval Postgraduate School
Monterey, California

3. Chairman, Code EC
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California

4. Chairman, MOVES
Department of Modeling, Virtual Environment and Simulation
Naval Postgraduate School
Monterey, California

5. Prof. Xiaoping Yun, Code EC/YX
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California

6. Prof. Fotis A. Papoulias, Code ME/Pa
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California

7. Military Instructor Joseph A. Sullivan
Department of Modeling, Virtual Environment and Simulation
Naval Postgraduate School
Monterey, California

8. Mr. David Leonardo Lencastre Sicuro
Rio de Janeiro, RJ – Cep: 20541-120
Brazil

9. Diretoria de Engenharia Naval
Rio de Janeiro, RJ – Cep: 20010-000
Brazil